

(19) 世界知的所有権機関  
国際事務局(43) 国際公開日  
2001年11月1日 (01.11.2001)

PCT

(10) 国際公開番号  
WO 01/82606 A1

(51) 国際特許分類: H04N 5/92, G11B 20/10

(21) 国際出願番号: PCT/JP01/03415

(22) 国際出願日: 2001年4月20日 (20.04.2001)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(30) 優先権データ:  
特願2000-183771 2000年4月21日 (21.04.2000) JP  
特願2000-271552 2000年9月7日 (07.09.2000) JP

(71) 出願人(米国を除く全ての指定国について): ソニー株式会社 (SONY CORPORATION) [JP/JP]; 〒141-0001 東京都品川区北品川6丁目7番35号 Tokyo (JP).

(72) 発明者; および  
(75) 発明者/出願人(米国についてのみ): 加藤元樹 (KATO, Motoki) [JP/JP]. 浜田俊也 (HAMADA, Toshiya) [JP/JP]; 〒141-0001 東京都品川区北品川6丁目7番35号 ソニー株式会社内 Tokyo (JP).

(74) 代理人: 小池 晃, 外 (KOIKE, Akira et al.); 〒105-0001 東京都港区虎ノ門二丁目6番4号 第11森ビル Tokyo (JP).

(81) 指定国(国内): AU, BR, CA, CN, CZ, HU, ID, IL, IN, KR, MX, NO, NZ, PL, RO, RU, SG, SK, US, VN, ZA.

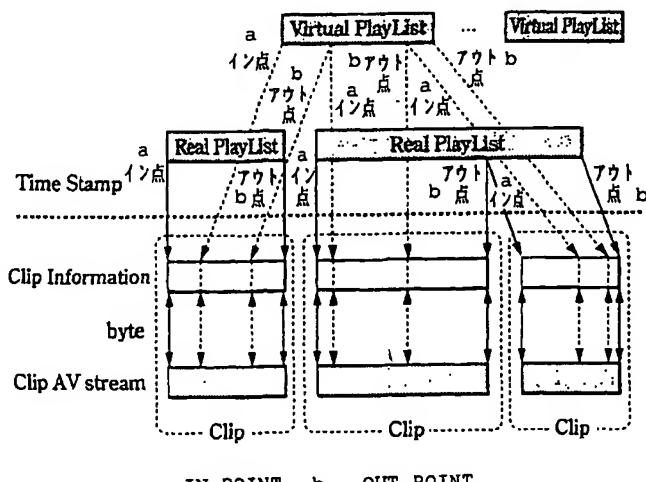
(84) 指定国(広域): ヨーロッパ特許 (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

添付公開書類:  
— 国際調査報告書  
補正書

2文字コード及び他の略語については、定期発行される各PCTガゼットの巻頭に掲載されている「コードと略語のガイドスノート」を参照。

(54) Title: INFORMATION PROCESSING APPARATUS AND METHOD, PROGRAM, AND RECORDED MEDIUM

(54) 発明の名称: 情報処理装置及び方法、プログラム、並びに記録媒体



(57) Abstract: A CPI-type is described in a Playlist(). The CPI-type includes an EP-map type and a TU-map type. If the position of an I picture can be analyzed, the EP-map type is used; and if not, the TU-map type is used. As a result, AV stream data recorded after analyzing the position of the I picture and AV stream data recorded without analyzing the position of the I picture can be commonly managed.

(57) 要約:

WO 01/82606 A1

Playlist()には、CPI\_typeが記述される。CPI\_typeには、EP\_map typeと、TU\_map typeがある。Iピクチャの位置が分析できる場合、EP\_map typeが用いられ、Iピクチャの位置が分析できない場合、TU\_map typeが用いられる。これにより、Iピクチャの位置を分析してAVストリームデータと、分析しないで記録するAVストリームデータとを共通に管理できるようにする。

## 明細書

## 情報処理装置及び方法、プログラム、並びに記録媒体

## 技術分野

本発明は情報処理装置、再生方法、プログラム、並びに記録媒体に関し、特に、G U I 等に説明表示する情報、主の再生経路の情報、副の再生経路の情報、主の再生経路を構成する個々の再生区間の間の接続情報、ユーザが所望したシーンにセットするブックマークやリジューム点の情報等の情報を含むファイルを記録する情報処理装置、再生方法、プログラム、並びに記録媒体に関する。

## 背景技術

近年、記録再生装置から取り外し可能なディスク型の記録媒体として、各種の光ディスクが提案されつつある。このような記録可能な光ディスクは、数ギガバイトの大容量メディアとして提案されており、ビデオ信号等の A V (Audio Visual) 信号を記録するメディアとしての期待が高い。この記録可能な光ディスクに記録するディジタルの A V 信号のソース（供給源）としては、C S ディジタル衛星放送や B S ディジタル放送があり、また、将来はディジタル方式の地上波テレビジョン放送等も提案されている。

ここで、これらのソースから供給されるディジタルビデオ信号は、通常 M P E G (Moving Picture Experts Group) 2 方式で画像圧縮されているのが一般的である。また、記録装置には、その装置固有の記録レートが定められている。従来の民生用映像蓄積メディアで、ディジタル放送由来のディジタルビデオ信号を記録する場合、アナログ記録方式であれば、ディジタルビデオ信号をデコード後、帯域制限をして記録する。或いは、M P E G 1 V i d e o 、 M P E G 2 V i d e o 、 D V 方式をはじめとするディジタル記録方式であれば、1 度デコードされた後に、その装置固有の記録レート・符号化方式で再エンコードされて記録される。

しかしながら、このような記録方法は、供給されたビットストリームを1度デコードし、その後で帯域制限や再エンコードを行って記録するため、画質の劣化を伴う。画像圧縮されたディジタル信号の記録をする場合、入力されたディジタル信号の伝送レートが記録再生装置の記録レートを超えない場合には、供給されたビットストリームをデコードや再エンコードすることなく、そのまま記録する方法が最も画質の劣化が少ない。但し、画像圧縮されたディジタル信号の伝送レートが記録媒体としてのディスクの記録レートを超える場合には、記録再生装置でデコード後、伝送レートがディスクの記録レートの上限以下になるように、再エンコードをして記録する必要はある。

また、入力ディジタル信号のビットレートが時間により増減する可変レート方式によって伝送されている場合には、回転ヘッドが固定回転数であるために記録レートが固定レートになるテープ記録方式に比べ、1度バッファにデータを蓄積し、バースト的に記録ができるディスク記録装置が記録媒体の容量をより無駄なく利用できる。

以上のように、デジタル放送が主流となる将来においては、データストリーマのように放送信号をデジタル信号のまま、デコードや再エンコードすることなく記録し、記録媒体としてディスクを使用した記録再生装置が求められると予測される。

ところで、上述したような記録装置により記録媒体にAVストリームデータを記録する場合、例えば、高速再生ができるようにするために、AVストリームデータを分析し、Iピクチャの位置を検出して、Iピクチャにアクセスできるようにして記録する場合と、AVストリームデータを分析せず、そのまま記録する場合とがある。

このような場合、従来、それぞれ専用のアプリケーションプログラムを用意し、それにより、AVストリームを、異なるフォーマットのAVストリーム（高速再生が可能なAVストリーム、又は不可能なAVストリーム）として記録媒体に記録するようになっていた。この結果、アプリケーションプログラムの開発に、費用と時間がかかる課題があった。また、それぞれのアプリケーションプログラムにより記録されたAVストリームは、異なるフォーマットのものとなるので、相

互の互換性がなくなり、共通の装置で再生することができなくなる課題があった。

更に、従来の記録装置では、例えば、オーディオデータを、所謂アフターレコードティングすることが困難である課題があった。

### 発明の開示

本発明の目的は、このような状況に鑑み、高速再生が可能なAVストリームと不可能なAVストリームを、共通に管理することにある。

また、本発明の他の目的は、アフターレコードティングを可能にすることにある。

本発明に係る情報処理装置は、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットのAVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランSPORTパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランSPORTパケットのAVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルを生成する第1の生成手段と、記録方法に応じて第1のテーブル又は第2のテーブルの一方を選択する選択手段と、選択されたテーブルをAVストリームデータとともに記録媒体に記録する第1の記録手段とを有する。

前記第1のテーブルは、EP\_mapであり、第2のテーブルは、TU\_mapとすることができる。

前記選択手段は、ノンコグニザント記録の際には、第2のテーブルを選択することができる。

前記選択手段は、セルフエンコード記録の際には、第1のテーブルを選択することができる。

前記選択手段は、コグニザント記録の際には、第1のテーブルを選択することができる。

前記AVストリームデータの再生を指定する再生指定情報を生成する第2の生成手段と、第2の生成手段により生成された再生指定情報を記録媒体に記録する第2の記録手段を更に有し、再生指定情報は、AVストリームデータの再生区間の時間情報を、プレゼンテーションタイムベースで表現するか、又はアライバル

タイムベースで表現するかを示す種別情報を含むようにすることができる。

前記AVストリームデータとともに第1のテーブルが記録されている場合、再生指定情報は、AVストリームデータの再生区間の時間情報を、プレゼンテーションタイムベースで表現し、AVストリームデータとともに第2のテーブルが記録されている場合、再生指定情報は、AVストリームデータの再生区間の時間情報を、アライバルタイムベースで表現することができる。

本発明に係る情報処理方法は、プレゼンテーションタイムスタンプと、それに応するアクセスユニットのAVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットのAVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルを生成する生成ステップと、記録方法に応じて第1のテーブル又は第2のテーブルの一方を選択する選択ステップと、選択されたテーブルをAVストリームデータとともに記録媒体に記録する記録ステップとを含む。

本発明に係る記録媒体のプログラムは、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットのAVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットのAVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルを生成する生成ステップと、記録方法に応じて第1のテーブル又は第2のテーブルの一方を選択する選択ステップと、選択されたテーブルをAVストリームデータと共に記録媒体に記録する記録ステップとを含む。

本発明に係るプログラムは、プレゼンテーションタイムスタンプと、これに対応するアクセスユニットのAVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットのAVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルを生成する生成ステップと、記録方法に応じて第1のテーブル又は第2のテーブルの一方を選択する選択ステップと、選択されたテーブルをAVストリームデータと共に記録

媒体に記録する記録ステップとをコンピュータに実行させる。

本発明に係る情報処理装置は、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットのAVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランsportパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランsportパケットのAVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている記録媒体から、第1のテーブル又は第2のテーブルの一方を再生する再生手段と、再生されたテーブルに基づいて、AVストリームデータの出力を制御する制御手段とを有する。

本発明に係る情報処理方法は、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットのAVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランsportパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランsportパケットのAVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている記録媒体から、第1のテーブル又は第2のテーブルの一方を再生する再生ステップと、再生されたテーブルに基づいて、AVストリームデータの出力を制御する制御手段とを含む。

本発明に係る記録媒体のプログラムは、プレゼンテーションタイムスタンプと、これに対応するアクセスユニットのAVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランsportパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランsportパケットのAVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている記録媒体から、第1のテーブル又は第2のテーブルの一方を再生する再生ステップと、再生されたテーブルに基づいて、AVストリームデータの出力を制御する制御手段とを含む。

本発明に係るプログラムは、プレゼンテーションタイムスタンプと、これに対応するアクセスユニットのAVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランsportパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランsportパケットのAVスト

リームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている記録媒体から、第1のテーブル又は第2のテーブルの一方を再生する再生ステップと、再生されたテーブルに基づいて、AVストリームデータの出力を制御する制御ステップとをコンピュータに実行させる。

本発明に係る記録媒体は、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットのAVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットのAVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている。

本発明に係る情報処理装置は、主の再生パスを示す第1の情報と、主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を生成する生成手段と、AVストリームデータと再生指定情報を記録媒体に記録する記録手段とを備える。

前記副の再生パスは、オーディオデータのアフターレコーディング用のパスとすることができる。

前記第1の情報は、Main\_pathであり、第2の情報は、Sub\_pathとすることができる。

前記第2の情報は、副の再生パスのタイプを表すタイプ情報、副の再生パスが参照するAVストリームのファイル名、副の再生パスのAVストリームのイン点とアウト点、及び再生パスのイン点が、主のパスの時間軸上で同期してスタートする主のパス上の時刻を含むようにすることができる。

本発明に係る情報処理方法は、主の再生パスを示す第1の情報と、主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を生成する生成ステップと、AVストリームデータと再生指定情報を記録媒体に記録する記録ステップとを含む。

本発明に係る記録媒体のプログラムは、主の再生パスを示す第1の情報と、主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を生成する生成ステップと、AVストリームデータと再生指定情

報を記録媒体に記録する記録ステップとを含む。

本発明に係るプログラムは、主の再生バスを示す第1の情報と、主の再生バスと同期して再生される副の再生バスを示す第2の情報により構成される再生指定情報を生成する生成ステップと、AVストリームデータと再生指定情報を記録媒体に記録する記録ステップとをコンピュータに実行させる。

前記情報処理装置は、主の再生バスを示す第1の情報と、前記主の再生バスと同期して再生される副の再生バスを示す第2の情報により構成される再生指定情報を、前記記録媒体から再生する再生手段と、再生された前記再生指定情報に基づいて、前記AVストリームデータの出力を制御する制御手段とを備える。

本発明に係る情報処理方法は、主の再生バスを示す第1の情報と、主の再生バスと同期して再生される副の再生バスを示す第2の情報により構成される再生指定情報を、記録媒体から再生する再生ステップと、再生された再生指定情報に基づいて、AVストリームデータの出力を制御する制御ステップとを含む。

本発明に係る記録媒体のプログラムは、主の再生バスを示す第1の情報と、主の再生バスと同期して再生される副の再生バスを示す第2の情報により構成される再生指定情報を、記録媒体から再生する再生ステップと、再生された再生指定情報に基づいて、AVストリームデータの出力を制御する制御ステップとを含む。

本発明に係るプログラムは、主の再生バスを示す第1の情報と、主の再生バスと同期して再生される副の再生バスを示す第2の情報により構成される再生指定情報を、記録媒体から再生する再生ステップと、再生された再生指定情報に基づいて、AVストリームデータの出力を制御する制御ステップとをコンピュータに実行させる。

本発明に係る記録媒体は、主の再生バスを示す第1の情報と、主の再生バスと同期して再生される副の再生バスを示す第2の情報により構成される再生指定情報が記録されている。

本発明に係る情報処理装置及び方法、記録媒体のプログラム、プログラム、並びに記録媒体においては、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランスポートパケットの到着時刻に基づいたア

ライバルタイムスタンプと、それに対応するトランSPORTパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録される。

本発明に係る情報処理装置及び方法、記録媒体のプログラム、並びにプログラムにおいては、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランSPORTパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランSPORTパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている記録媒体から、そのテーブルが再生され、それに基づいて、出力が制御される。

本発明に係る情報処理装置及び方法、記録媒体のプログラム、プログラム、並びに第2の記録媒体においては、主の再生バスを示す第1の情報と、前記主の再生バスと同期して再生される副の再生バスを示す第2の情報により構成される再生指定情報が記録される。

本発明に係る情報処理装置及び方法、記録媒体のプログラム、並びにプログラムにおいては、主の再生バスを示す第1の情報と、前記主の再生バスと同期して再生される副の再生バスを示す第2の情報により構成される再生指定情報が記録媒体から再生され、それに基づいて、出力が制御される。

本発明の更に他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。

#### 図面の簡単な説明

図1は、本発明を適用した記録再生装置の構成を示す図である。

図2は、記録再生装置により記録媒体に記録されるデータのフォーマットについて説明する図である。

図3は、Real PlayListとVirtual PlayListについて説明する図である。

図4A～図4Cは、Real PlayListの作成について説明する図である。

図5A～図5Cは、Real PlayListの削除について説明する図である。

図6A及び図6Bは、アセンブル編集について説明する図である。

図7は、Virtual PlayListにサブバスを設ける場合について説明する図である。

図8は、PlayListの再生順序の変更について説明する図である。

図9は、PlayList上のマークとClip上のマークについて説明する図である。

図10は、メニュー サムネイルについて説明する図である。

図11は、PlayListに付加されるマークについて説明する図である。

図12は、クリップに付加されるマークについて説明する図である。

図13は、PlayList、Clip、サムネイルファイルの関係について説明する図である。

図14は、ディレクトリ構造について説明する図である。

図15は、info.dvrのシンタクスを示す図である。

図16は、DVR volumeのシンタクスを示す図である。

図17は、Resumevolumeのシンタクスを示す図である。

図18は、UIAppInfovolumeのシンタクスを示す図である。

図19は、Character set valueのテーブルを示す図である。

図20は、TableOfPlayListのシンタクスを示す図である。

図21は、TableOfPlayListの他のシンタクスを示す図である。

図22は、MakersPrivateDataのシンタクスを示す図である。

図23は、xxxxx.rplsとyyyyy.vplsのシンタクスを示す図である。

図24A～図24Cは、PlayListについて説明する図である。

図25は、PlayListのシンタクスを示す図である。

図26は、PlayList\_typeのテーブルを示す図である。

図27は、UIAppinfoPlayListのシンタクスを示す図である。

図28A～図28Cは、図27に示したUIAppinfoPlayListのシンタクス内のフラグについて説明する図である。

図29は、PlayItemについて説明する図である。

図30は、PlayItemについて説明する図である。

図31は、PlayItemについて説明する図である。

図32は、PlayItemのシンタクスを示す図である。

図33は、IN\_timeについて説明する図である。

図34は、OUT\_timeについて説明する図である。

図35は、Connection\_Conditionのテーブルを示す図である。

図36A～図36Dは、Connection\_Conditionについて説明する図である。

図37は、BridgeSequenceInfoを説明する図である。

図38は、BridgeSequenceInfoのシンタクスを示す図である。

図39は、SubPlayItemについて説明する図である。

図40は、SubPlayItemのシンタクスを示す図である。

図41は、SubPath\_typeのテーブルを示す図である。

図42は、PlayListMarkのシンタクスを示す図である。

図43は、Mark\_typeのテーブルを示す図である。

図44は、Mark\_time\_stampを説明する図である。

図45は、zzzz.clipのシンタクスを示す図である。

図46は、ClipInfoのシンタクスを示す図である。

図47は、Clip\_stream\_typeのテーブルを示す図である。

図48は、offset\_SPNについて説明する図である。

図49は、offset\_SPNについて説明する図である。

図50A及び図50Bは、STC区間について説明する図である。

図51は、STC\_Infoについて説明する図である。

図52は、STC\_Infoのシンタクスを示す図である。

図53は、ProgramInfoを説明する図である。

図54は、ProgramInfoのシンタクスを示す図である。

図55は、VideoCodingInfoのシンタクスを示す図である。

図56は、Video\_formatのテーブルを示す図である。

図57は、frame\_rateのテーブルを示す図である。

図58は、display\_aspect\_ratioのテーブルを示す図である。

図59は、AudioCodingInfoのシンタクスを示す図である。

図60は、audio\_codingのテーブルを示す図である。

図 6 1 は、 `audio_component_type` のテーブルを示す図である。

図 6 2 は、 `sampling_frequency` のテーブルを示す図である。

図 6 3 は、 CPI について説明する図である。

図 6 4 は、 CPI について説明する図である。

図 6 5 は、 CPI のシンタクスを示す図である。

図 6 6 は、 CPI\_type のテーブルを示す図である。

図 6 7 は、 ビデオ EP\_map について説明する図である。

図 6 8 は、 EP\_map について説明する図である。

図 6 9 は、 EP\_map について説明する図である。

図 7 0 は、 EP\_map のシンタクスを示す図である。

図 7 1 は、 EP\_type values のテーブルを示す図である。

図 7 2 は、 EP\_map\_for\_one\_stream\_PID のシンタクスを示す図である。

図 7 3 は、 TU\_map について説明する図である。

図 7 4 は、 TU\_map のシンタクスを示す図である。

図 7 5 は、 ClipMark のシンタクスを示す図である。

図 7 6 は、 mark\_type のテーブルを示す図である。

図 7 7 は、 mark\_type\_stamp のテーブルを示す図である。

図 7 8 は、 menu.thmb と mark.thmb のシンタクスを示す図である。

図 7 9 は、 Thumbnail のシンタクスを示す図である。

図 8 0 は、 thumbnail\_picture\_format のテーブルを示す図である。

図 8 1 A 及び図 8 1 B は、 tn\_block について説明する図である。

図 8 2 は、 DVR MPEG 2 のトランスポートストリームの構造について説明する図である。

図 8 3 は、 DVR MPEG 2 のトランスポートストリームのレコーダモデルを示す図である。

図 8 4 は、 DVR MPEG 2 のトランスポートストリームのプレーヤモデルを示す図である。

図 8 5 は、 source\_packet のシンタクスを示す図である。

図 8 6 は、 TP\_extra\_header のシンタクスを示す図である。

図87は、copy permission indicatorのテーブルを示す図である。

図88は、シームレス接続について説明する図である。

図89は、シームレス接続について説明する図である。

図90は、シームレス接続について説明する図である

図91は、シームレス接続について説明する図である。

図92は、シームレス接続について説明する図である

図93は、オーディオのオーバーラップについて説明する図である。

図94は、BridgeSequenceを用いたシームレス接続について説明する図である。

図95は、BridgeSequenceを用いないシームレス接続について説明する図である。

図96は、DVR STDモデルを示す図である。

図97は、復号、表示のタイミングチャートである。

図98は、PlayListファイルのシンタクスを示す図である。

図99は、図98のPlayListファイル中のUIAppInfoPlayListのシンタクスを示す図である。

図100は、図98のPlayListファイル中のPlayList()のシンタクスを示す図である。

図101は、SubPlayItemのシンタクスを示す図である。

図102は、Real PlayListの作成方法を説明するフローチャートである。

図103は、Virtual PlayListの作成方法を説明するフローチャートである。

図104は、PlayListの再生方法を説明するフローチャートである。

図105は、PlayListのSubバスの再生方法を説明するフローチャートである。

図106は、PlayListMarkの作成方法を説明するフローチャートである。

図107は、PlayListMarkを使用した頭出し再生方法を説明するフローチャートである。

図108は、媒体を説明する図である。

発明を実施するための最良の形態

以下に、本発明が適用された情報処理装置及び方法、プログラム、並びに記録媒体について、図面を参照して説明する。図1は、本発明を適用した記録再生装置1の内部構成例を示す図である。先ず、外部から入力された信号を記録媒体に記録する動作を行う部分の構成について説明する。記録再生装置1は、アナログデータ、又は、デジタルデータを入力し、記録することができる構成とされている。

端子11には、アナログのビデオ信号が、端子12には、アナログのオーディオ信号が、それぞれ入力される。端子11に入力されたビデオ信号は、解析部14とAVエンコーダ15に、それぞれ出力される。端子12に入力されたオーディオ信号は、AVエンコーダ15に出力される。解析部14は、入力されたビデオ信号からシーンチェンジ等の特徴点を抽出する。

AVエンコーダ15は、入力されたビデオ信号とオーディオ信号を、それぞれ符号化し、符号化ビデオストリーム(V)、符号化オーディオストリーム(A)、及びAV同期等のシステム情報(S)をマルチプレクサ16に出力する。

符号化ビデオストリームは、例えば、MPEG(Moving Picture Expert Group)2方式により符号化されたビデオストリームであり、符号化オーディオストリームは、例えば、MPEG1方式により符号化されたオーディオストリームや、ドルビーAC3方式により符号化されたオーディオストリーム等である。マルチプレクサ16は、入力されたビデオ及びオーディオのストリームを、入力システム情報に基づいて多重化して、スイッチ17を介して多重化ストリーム解析部18とソースパケットタイザ19に出力する。

多重化ストリームは、例えば、MPEG2トランSPORTストリームやMPEG2プログラムストリームである。ソースパケットタイザ19は、入力された多重化ストリームを、このストリームを記録させる記録媒体100のアプリケーションフォーマットに従って、ソースパケットから構成されるAVストリームを符号化する。AVストリームは、ECC(誤り訂正)符号化部20、変調部21で所定の処理が施され、書込部22に出力される。書込部22は、制御部23から出力される制御信号に基づいて、記録媒体100にAVストリームファイルを書き込む(記録する)。

デジタルインタフェース又はデジタルテレビジョンチューナから入力されるデジタルテレビジョン放送等のトランスポートストリームは、端子13に入力される。端子13に入力されたトランスポートストリームの記録方式には、2通りあり、これらは、トランスペアレントに記録する方式と、記録ビットレートを下げる等の目的のために再エンコードをした後に記録する方式である。記録方式の指示情報は、ユーザインタフェースとしての端子24から制御部23へ入力される。

入力トランスポートストリームをトランスペアレントに記録する場合、端子13に入力されたトランスポートストリームは、多重化ストリーム解析部18と、ソースパケットタイザ19に出力される。これ以降の記録媒体100へAVストリームが記録されるまでの処理は、上述の入力オーディオ浸透とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

入力トランスポートストリームを再エンコードした後に記録する場合、端子13に入力されたトランスポートストリームは、デマルチブレクサ26に入力される。デマルチブレクサ26は、入力されたトランスポートストリームに対してデマルチブレクス処理を施し、ビデオストリーム(V)、オーディオストリーム(A)、及びシステム情報(S)を抽出する。

デマルチブレクサ26により抽出されたストリーム(情報)のうち、ビデオストリームはAVデコーダ27に、オーディオストリームとシステム情報はマルチブレクサ16に、それぞれ出力される。AVデコーダ27は、入力されたビデオストリームを復号し、その再生ビデオ信号をAVエンコーダ15に出力する。AVエンコーダ15は、入力ビデオ信号を符号化し、符号化ビデオストリーム(V)をマルチブレクサ16に出力する。

一方、デマルチブレクサ26から出力され、マルチブレクサ16に入力されたオーディオストリームとシステム情報、及び、AVエンコーダ15から出力されたビデオストリームは、入力システム情報に基づいて、多重化されて、多重化ストリームとして多重化ストリーム解析部18とソースパケットタイザ19にスイッチ17を介して出力される。これ以後の記録媒体100へAVストリームが記録されるまでの処理は、上述の入力オーディオ信号とビデオ信号を符号化して記

録する場合と同一の処理なので、その説明は省略する。

記録再生装置1は、AVストリームのファイルを記録媒体100に記録すると共に、そのファイルを説明するアプリケーションデータベース情報も記録する。アプリケーションデータベース情報は、制御部23により作成される。制御部23への入力情報は、解析部14からの動画像の特徴情報、多重化ストリーム解析部18からのAVストリームの特徴情報、及び端子24から入力されるユーザからの指示情報である。

解析部14から供給される動画像の特徴情報は、入力動画像信号の中の特徴的な画像に関する情報であり、例えば、プログラムの開始点、シーンチェンジ点、コマーシャル(CM)の開始・終了点等の指定情報(マーク)であり、また、その指定場所の画像のサムネイル画像の情報も含まれる。

多重化ストリーム解析部18からのAVストリームの特徴情報は、記録されるAVストリームの符号化情報に関する情報であり、例えば、AVストリーム内のIピクチャのアドレス情報、AVストリームの符号化パラメータ、AVストリームの中の符号化パラメータの変化点情報、ビデオストリームの中の特徴的な画像に関する情報(マーク)等である。

端子24からのユーザの指示情報は、AVストリームの中の、ユーザが指定した再生区間の指定情報、その再生区間の内容を説明するキャラクター文字、ユーザが好みのシーンにセットするブックマークやリピューム点の情報等である。

制御部23は、上記の入力情報に基づいて、AVストリームのデータベース(Clip)、AVストリームの再生区間(PlayItem)をグループ化したもの(Playlist)のデータベース、記録媒体100の記録内容の管理情報(info.dvr)、及びサムネイル画像の情報を作成する。これらの情報から構成されるアプリケーションデータベース情報は、AVストリームと同様にして、ECC符号化部20、変調部21で処理されて、書込部22へ入力される。書込部22は、制御部23から出力される制御信号に基づいて、記録媒体100へデータベースファイルを記録する。

上述したアプリケーションデータベース情報についての詳細は後述する。

このようにして記録媒体100に記録されたAVストリームファイル(画像データと音声データのファイル)と、アプリケーションデータベース情報が再生さ

れる場合、先ず、制御部23は、読出部28に対して、記録媒体100からアプリケーションデータベース情報を読み出すように指示する。そして、読出部28は、記録媒体100からアプリケーションデータベース情報を読み出し、そのアプリケーションデータベース情報は、復調部29、ECC復号部30の処理を経て、制御部23へ入力される。

制御部23は、アプリケーションデータベース情報に基づいて、記録媒体100に記録されているPlayListの一覧を端子24のユーザインタフェースへ出力する。ユーザは、PlayListの一覧から再生したいPlayListを選択し、再生を指定されたPlayListに関する情報が制御部23へ入力される。制御部23は、そのPlayListの再生に必要なAVストリームファイルの読み出しを、読出部28に指示する。読出部28は、その指示に従い、記録媒体100から対応するAVストリームを読み出し復調部29に出力する。復調部29に入力されたAVストリームは、所定の処理が施されることにより復調され、更にECC復号部30の処理を経て、ソースデパケッタイザ31出力される。

ソースデパケッタイザ31は、記録媒体100から読み出され、所定の処理が施されたアプリケーションフォーマットのAVストリームを、デマルチブレクサ26に出力できるストリームに変換する。デマルチブレクサ26は、制御部23により指定されたAVストリームの再生区間(PlayItem)を構成するビデオストリーム(V)、オーディオストリーム(A)、及びAV同期等のシステム情報(S)を、AVデコーダ27に出力する。AVデコーダ27は、ビデオストリームとオーディオストリームを復号し、再生ビデオ信号と再生オーディオ信号を、それぞれ対応する端子32と端子33から出力する。

また、ユーザインタフェースとしての端子24から、ランダムアクセス再生や特殊再生を指示する情報が入力された場合、制御部23は、AVストリームのデータベース(Clip)の内容に基づいて、記憶媒体100からのAVストリームの読み出位置を決定し、そのAVストリームの読み出しを、読出部28に指示する。例えば、ユーザにより選択されたPlayListを、所定の時刻から再生する場合、制御部23は、指定された時刻に最も近いタイムスタンプを持つIピクチャからのデータを読み出すように読出部28に指示する。

また、ユーザによって高速再生(Fast-forward playback)が指示された場合、制御部23は、AVストリームのデータベース(Clip)に基づいて、AVストリームの中のI-ピクチャデータを順次連続して読み出すように読み出部28に指示する。

読み出部28は、指定されたランダムアクセスポイントからAVストリームのデータを読み出し、読み出されたデータは、後段の各部の処理を経て再生される。

次に、ユーザが、記録媒体100に記録されているAVストリームの編集をする場合を説明する。ユーザが、記録媒体100に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合、例えば、番組Aという歌番組から歌手Aの部分を再生し、その後続けて、番組Bという歌番組の歌手Aの部分を再生したいといった再生経路を作成したい場合、ユーザインタフェースとしての端子24から再生区間の開始点(イン点)と終了点(アウト点)の情報が制御部23に入力される。制御部23は、AVストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベースを作成する。

ユーザが、記録媒体100に記録されているAVストリームの一部を消去したい場合、ユーザインタフェースとしての端子24から消去区間のイン点とアウト点の情報が制御部23に入力される。制御部23は、必要なAVストリーム部分だけを参照するようにPlayListのデータベースを変更する。また、AVストリームの不必要的ストリーム部分を消去するように、書込部22に指示する。

ユーザが、記録媒体100に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合であり、且つ、それぞれの再生区間をシームレスに接続したい場合について説明する。このような場合、制御部23は、AVストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベースを作成し、更に、再生区間の接続点付近のビデオストリームの部分的な再エンコードと再多重化を行う。

先ず、端子24から再生区間のイン点のピクチャの情報と、アウト点のピクチャの情報が制御部23へ入力される。制御部23は、読み出部28にイン点側ピクチャとアウト点側のピクチャを再生するために必要なデータの読み出しを指示する。そして、読み出部28は、記録媒体100からデータを読み出し、そのデータは、復調部29、ECC復号部30、ソースデパケットライザ31を経て、デマル

チプレクサ26に出力される。

制御部23は、デマルチプレクサ26に入力されたデータを解析して、ビデオストリームの再エンコード方法 (picture\_coding\_typeの変更、再エンコードする符号化ビット量の割り当て) と、再多重化方式を決定し、その方式をAVエンコーダ15とマルチプレクサ16に供給する。

次に、デマルチプレクサ26は、入力されたストリームをビデオストリーム (V)、オーディオストリーム (A)、及びシステム情報 (S) に分離する。ビデオストリームは、「AVデコーダ27に入力されるデータ」と「マルチプレクサ16に入力されるデータ」がある。前者のデータは、再エンコードするために必要なデータであり、これはAVデコーダ27で復号され、復号されたピクチャはAVエンコーダ15で再エンコードされて、ビデオストリームにされる。後者のデータは、再エンコードをしないで、オリジナルのストリームからコピーされるデータである。オーディオストリーム、システム情報については、直接、マルチプレクサ16に入力される。

マルチプレクサ16は、制御部23から入力された情報に基づいて、入力ストリームを多重化し、多重化ストリームを出力する。多重化ストリームは、ECC符号化部20、変調部21で処理されて、書込部22に入力される。書込部22は、制御部23から供給される制御信号に基づいて、記録媒体100にAVストリームを記録する。

以下、アプリケーションデータベース情報や、その情報に基づく再生、編集といった操作に関する説明をする。図2は、アプリケーションフォーマットの構造を説明する図である。アプリケーションフォーマットは、AVストリームの管理のためにPlayListとClipの2つのレイヤをもつ。Volume Informationは、ディスク内の全てのClipとPlayListの管理をする。ここでは、1つのAVストリームとその付属情報のペアを1つのオブジェクトとし、これをClipという。AVストリームファイルはClip AV stream fileといい、この付属情報は、Clip Information fileという。

1つのClip AV stream fileは、MPEG2トランスポートストリームをアプリケーションフォーマットによって規定される構造に配置したデータをストアする。

一般的に、ファイルは、バイト列として扱われるが、Clip AV stream fileのコンテンツは、時間軸上に展開され、Clipの中のエントリポイントは、主に時間ベースで指定される。所定のClipへのアクセスポイントのタイムスタンプが与えられたとき、Clip Information fileは、Clip AV stream fileの中でデータの読み出しを開始すべきアドレス情報を見つけるために役立つ。

PlayListについて、図3を参照して説明する。PlayListは、Clipの中からユーザが見たい再生区間を選択し、これを簡単に編集することができるようするために設けられている。1つのPlayListは、Clipの中の再生区間の集まりである。所定のClipの中の1つの再生区間は、PlayItemと呼ばれ、それは、時間軸上のイン点 (IN) とアウト点 (OUT) の対で表される。したがって、PlayListは、複数のPlayItemが集まることにより構成される。

PlayListには、2つのタイプがある。1つは、Real PlayListであり、もう1つは、Virtual PlayListである。Real PlayListは、それが参照しているClipのストリーム部分を共有している。すなわち、Real PlayListは、その参照しているClipのストリーム部分に相当するデータ容量をディスクの中で占め、Real PlayListが消去された場合、それが参照しているClipのストリーム部分もまたデータが消去される。

Virtual PlayListは、Clipのデータを共有していない。したがって、Virtual PlayListが変更又は消去されたとしても、Clipの内容には何も変化が生じない。

次に、Real PlayListの編集について説明する。図4 Aは、Real PlayListのクリエイト(create:作成)に関する図であり、AVストリームが新しいClipとして記録される場合、そのClip全体を参照するReal PlayListが新たに作成される操作である。

図4 Bは、Real PlayListのディバイド(divide:分割)に関する図であり、Real PlayListが所望な点で分けられて、2つのReal PlayListに分割される操作である。この分割という操作は、例えば、1つのPlayListにより管理される1つのクリップ内に、2つの番組が管理されているような場合に、ユーザが1つ1つの番組として登録(記録)し直したいといったようなときに行われる。この操作により、Clipの内容が変更される(Clip自体が分割される)ことはない。

図4 Cは、Real PlayListのコンパイン(combine：結合)に関する図であり、2つのReal PlayListを結合して、1つの新しいReal PlayListにする操作である。この結合という操作は、例えば、ユーザが2つの番組を1つの番組として登録し直したいといったようなときに行われる。この操作により、Clipが変更される(Clip自体が1つにされる)ことはない。

図5 Aは、Real PlayList全体のデリート(delete：削除)に関する図であり、所定のReal PlayList全体を消去する操作がされた場合、削除されたReal PlayListが参照するClipの、対応するストリーム部分も削除される。

図5 Bは、Real PlayListの部分的な削除に関する図であり、Real PlayListの所望な部分が削除された場合、対応するPlayItemが、必要なClipのストリーム部分だけを参照するように変更される。そして、Clipの対応するストリーム部分は削除される。

図5 Cは、Real PlayListのミニマイズ(Minimize：最小化)に関する図であり、Real PlayListに対応するPlayItemを、Virtual PlayListに必要なClipのストリーム部分だけを参照するようにする操作である。Virtual PlayListにとって不必要的Clipの、対応するストリーム部分は削除される。

上述したような操作により、Real PlayListが変更されて、そのReal PlayListが参照するClipのストリーム部分が削除された場合、その削除されたClipを使用しているVirtual PlayListが存在し、そのVirtual PlayListにおいて、削除されたClipにより問題が生じる可能性がある。

そのようなことが生じないように、ユーザに、削除という操作に対して、「そのReal PlayListが参照しているClipのストリーム部分を参照しているVirtual PlayListが存在し、もし、そのReal PlayListが消去されると、そのVirtual PlayListもまた消去されることになるが、それでも良いか？」といったメッセージ等を表示させることにより、確認(警告)を促した後に、ユーザの指示により削除の処理を実行、又は、キャンセルする。又は、Virtual PlayListを削除する代わりに、Real PlayListに対してミニマイズの操作が行われるようにする。

次にVirtual PlayListに対する操作について説明する。Virtual PlayListに対して操作が行われたとしても、Clipの内容が変更されることはない。図6 A及び

図 6 B は、アセンブル(Assemble) 編集 (IN-OUT 編集)に関する図であり、ユーザが見たいと

所望した再生区間のPlayItemを作り、Virtual PlayListを作成するといった操作である。PlayItem間のシームレス接続が、アプリケーションフォーマットによりサポートされている（後述）。

図 6 A に示したように、2つのReal PlayList 1, 2 と、それぞれのReal PlayListに対応するClip 1, 2 が存在している場合に、ユーザがReal PlayList 1 内の所定の区間 (In 1 乃至 Out 1 までの区間 : PlayItem 1) を再生区間として指示し、続けて再生する区間として、Real PlayList 2 内の所定の区間 (In 2 乃至 Out 2 までの区間 : PlayItem 2) を再生区間として指示したとき、図 6 B に示すように、PlayItem 1 と PlayItem 2 から構成される 1 つのVirtual PlayListが作成される。

次に、Virtual PlayList の再編集(Re-editing)について説明する。再編集には、Virtual PlayListの中のイン点やアウト点の変更、Virtual PlayListへの新しいPlayItemの挿入(insert)や追加 append)、Virtual PlayListの中のPlayItemの削除等がある。また、Virtual PlayListそのものを削除することもできる。

図 7 は、Virtual PlayListへのオーディオのアフレコ(Audio dubbing (post recording))に関する図であり、Virtual PlayListへのオーディオのアフレコをサブバスとして登録することである。このオーディオのアフレコは、アプリケーションフォーマットによりサポートされている。Virtual PlayListのメインバスのAVストリームに、付加的なオーディオストリームが、サブバスとして付加される。

Real PlayListとVirtual PlayListで共通の操作として、図 8 に示すようなPlayListの再生順序の変更(Moving)がある。この操作は、ディスク(ボリューム)の中でのPlayListの再生順序の変更であり、アプリケーションフォーマットにおいて定義されるTable Of PlayList (図 20 等を参照して後述する) によってサポートされる。この操作により、Clipの内容が変更されるようなことはない。

次に、マーク (Mark) について説明する。マークは、Clip及びPlayListの中のハイライトや特徴的な時間を指定するために設けられている。Clipに付加される

マークは、AVストリームの内容に起因する特徴的なシーンを指定する、例えば、シーンチェンジ点等である。PlayListを再生するとき、そのPlayListが参照するClipのマークを参照して、使用することができる。

PlayListに付加されるマークは、主にユーザによってセットされる、例えば、ブックマークやリリューム点等である。Clip又はPlayListにマークをセットすることは、マークの時刻を示すタイムスタンプをマークリストに追加することにより行われる。また、マークを削除することは、マークリストの中から、そのマークのタイムスタンプを除去することである。したがって、マークの設定や削除により、AVストリームは何の変更もされない。

次に、サムネイルについて説明する。サムネイルは、Volume、PlayList、及びClipに付加される静止画である。サムネイルには、2つの種類があり、1つは、内容を表す代表画としてのサムネイルである。これは主としてユーザがカーソル（不図示）等を操作して見たいものを選択するためのメニュー画面で使われるものである。もう1つは、マークが指しているシーンを表す画像である。

Volumeと各Playlistは代表画を持つことができるようとする必要がある。Volumeの代表画は、ディスク（記録媒体100、以下、記録媒体100はディスク状のものであるとし、適宜、ディスクと記述する）を記録再生装置1の所定の場所にセットしたときに、そのディスクの内容を表す静止画を最初に表示する場合等に用いられる想定している。Playlistの代表画は、Playlistを選択するメニュー画面において、Playlistの内容を表すための静止画として用いられる想定している。

Playlistの代表画として、Playlistの最初の画像をサムネイル（代表画）にすることが考えられるが、必ずしも再生時刻0の先頭の画像が内容を表す上で最適な画像とは限らない。そこで、Playlistのサムネイルとして、任意の画像をユーザが設定できるようとする。以上2種類のサムネイルをメニューサムネイルという。メニューサムネイルは頻繁に表示されるため、ディスクから高速に読み出される必要がある。このため、全てのメニューサムネイルを1つのファイルに格納することが効率的である。メニューサムネイルは、必ずしもボリューム内の動画から抜き出したピクチャである必要はなく、図10に示すように、パーソナルコ

ンピュータやディジタルスチルカメラから取り込まれた画像でもよい。

一方、ClipとPlaylistには、複数個のマークを打てる必要があり、マーク位置の内容を知るためにマーク点の画像を容易に見ることができるようにする必要がある。このようなマーク点を表すピクチャをマークサムネイル (Mark Thumbnail) という。したがって、サムネイルの元となる画像は、外部から取り込んだ画像よりも、マーク点の画像を抜き出したものが主となる。

図11は、PlayListに付けられるマークと、そのマークサムネイルの関係について示す図であり、図12は、Clipに付けられるマークと、そのマークサムネイルの関係について示す図である。マークサムネイルは、メニューサムネイルと異なり、Playlistの詳細を表すときに、サブメニュー等で使われるため、短いアクセス時間で読み出されるようなことは要求されない。そのため、サムネイルが必要になる度に、記録再生装置1がファイルを開き、そのファイルの一部を読み出すことで多少時間がかかるても、問題にはならない。

また、ボリューム内に存在するファイル数を減らすために、全てのマークサムネイルは1つのファイルに格納するのがよい。Playlistはメニューサムネイル1つと複数のマークサムネイルを有することができるが、Clipは直接ユーザが選択する必要性がない（通常、Playlist経由で指定する）ため、メニューサムネイルを設ける必要はない。

図13は、上述したことを考慮した場合のメニューサムネイル、マークサムネイル、PlayList、及びClipの関係について示した図である。メニューサムネイルファイルには、PlayList毎に設けられたメニューサムネイルがファイルされている。メニューサムネイルファイルには、ディスクに記録されているデータの内容を代表するボリュームサムネイルが含まれている。マークサムネイルファイルは、各PlayList毎と各Clip毎に作成されたサムネイルがファイルされている。

次に、CPI (Characteristic Point Information) について説明する。CPIは、Clipインフォメーションファイルに含まれるデータであり、主に、それはClipへのアクセスポイントのタイムスタンプが与えられたとき、Clip AV stream fileの中でデータの読み出しを開始すべきデータアドレスを見つけるために用いられる。本実施の形態では、2種類のCPIを用いる。1つは、EP\_mapであり、

もう1つは、TU\_mapである。

EP\_mapは、エントリポイント（EP）データのリストであり、これは、エレメンタリーストリーム及びトランスポートストリームから抽出されたものである。これは、AVストリームの中でデコードを開始すべきエントリポイントの場所を見つけるためのアドレス情報を持つ。1つのEPデータは、プレゼンテーションタイムスタンプ（PTS）と、そのPTSに対応するアクセスユニットのAVストリームの中のデータアドレスの対で構成される。

EP\_mapは、主に2つの目的のために使用される。第1に、PlayListの中でプレゼンテーションタイムスタンプによって参照されるアクセスユニットのAVストリームの中のデータアドレスを見つけるために使用される。第2に、ファーストフォワード再生やファーストリバース再生のために使用される。記録再生装置1が、入力AVストリームを記録する場合、そのストリームのシンタクスを解析することができるとき、EP\_mapが作成され、ディスクに記録される。

TU\_mapは、デジタルインターフェースを通して入力されるトランスポートパケットの到着時刻に基づいたタイムユニット（TU）データのリストを持つ。これは、到着時刻ベースの時間とAVストリームの中のデータアドレスとの関係を与える。記録再生装置1が、入力AVストリームを記録する場合、そのストリームのシンタクスを解析することができないとき、TU\_mapが作成され、ディスクに記録される。

STCInfoは、MPEG2トランスポートストリームをストアしているAVストリームファイルの中にあるSTCの不連続点情報をストアする。AVストリームがSTCの不連続点を持つ場合、そのAVストリームファイルの中で同じ値のPTSが現れるかもしれない。そのため、AVストリーム上のある時刻をPTSベースで指す場合、アクセスポイントのPTSだけではそのポイントを特定するためには不十分である。更に、そのPTSを含むところの連続なSTC区間のインデックスが必要である。連続なSTC区間を、このフォーマットではSTC-sequenceと呼び、そのインデックスをSTC-sequence-idと呼ぶ。STC-sequenceの情報は、Clip Information fileのSTCInfoで定義される。STC-sequence-idは、EP\_mapを持つAVストリームファイルで使用するものであり、TU\_mapを持つAVストリーム

ファイルではオプションである。

プログラムは、エレメンタリストリームの集まりであり、これらのストリームの同期再生のために、ただ1つのシステムタイムベースを共有するものである。再生装置（図1の記録再生装置1）にとって、AVストリームのデコードに先立ち、そのAVストリームの内容がわかるることは有用である。例えば、ビデオやオーディオのエレメンタリーストリームを伝送するトランSPORTパケットのPIDの値や、ビデオやオーディオのコンポーネント種類（例えば、HDTVのビデオとMPEG-2 AACのオーディオストリーム等）等の情報である。この情報はAVストリームを参照するところのPlayListの内容をユーザに説明するところのメニュー画面を作成するのに有用であるし、また、AVストリームのデコードに先立って、再生装置のAVデコーダ及びデマルチプレクサの初期状態をセットするために役立つ。この理由のために、Clip Information fileは、プログラムの内容を説明するためのProgramInfoを持つ。

MPEG2トランSPORTストリームをストアしているAVストリームファイルは、ファイルの中でプログラム内容が変化するかもしれない。例えば、ビデオエレメンタリーストリームを伝送するところのトランSPORTパケットのPIDが変化したり、ビデオストリームのコンポーネント種類がSDTVからHDTVに変化する等である。

ProgramInfoは、AVストリームファイルの中でのプログラム内容の変化点の情報をストアする。AVストリームファイルの中で、このフォーマットで定めるところのプログラム内容が一定である区間をProgram-sequenceという。Program-sequenceは、EP\_mapを持つAVストリームファイルで使用するものであり、TU\_mapを持つAVストリームファイルではオプションである。

ここでは、セルフエンコードのストリームフォーマット（SESF）を定義する。SESFは、アナログ入力信号を符号化する目的、及びデジタル入力信号（例えばDV）をデコードしてからMPEG2トランSPORTストリームに符号化する場合に用いられる。

SESFは、MPEG-2トランSPORTストリーム及びAVストリームについてのエレメンタリーストリームの符号化制限を定義する。記録再生装置1が、

S E S Fストリームをエンコードし、記録する場合、EP\_mapが作成され、ディスクに記録される。

デジタル放送のストリームは、次に示す方式のうちのいずれかが用いられて記録媒体100に記録される。先ず、デジタル放送のストリームをS E S Fストリームにトランスコーディングする。この場合、記録されたストリームは、S E S Fに準拠しなければならない。この場合、EP\_mapが作成されて、ディスクに記録されなければならない。

或いは、デジタル放送ストリームを構成するエレメンタリーストリームを新しいエレメンタリーストリームにトランスコーディングし、そのデジタル放送ストリームの規格化組織が定めるストリームフォーマットに準拠した新しいトランスポートストリームに再多重化する。この場合、EP\_mapが作成されて、ディスクに記録されなければならない。

例えば、入力ストリームがI S D B（日本のデジタルB S放送の規格名称）準拠のM P E G-2トランスポートストリームであり、それがH D T VビデオストリームとM P E G A A Cオーディオストリームを含むとする。H D T VビデオストリームをS D T Vビデオストリームにトランスコーディングし、そのS D T VビデオストリームとオリジナルのA A CオーディオストリームをT Sに再多重化する。S D T Vストリームと記録されるトランスポートストリームは、共にI S D Bフォーマットに準拠しなければならない。

デジタル放送のストリームが、記録媒体100に記録される際の他の方式として、入力トランスポートストリームをトランスペアレントに記録する（入力トランスポートストリームを何も変更しないで記録する）場合であり、そのときにEP\_mapが作成されてディスクに記録される。

又は、入力トランスポートストリームをトランスペアレントに記録する（入力トランスポートストリームを何も変更しないで記録する）場合であり、そのときにTU\_mapが作成されてディスクに記録される。

次にディレクトリとファイルについて説明する。以下、記録再生装置1をD V R（Digital Video Recording）と適宜記述する。図14はディスク上のディレクトリ構造の一例を示す図である。D V Rのディスク上に必要なディレクトリは、

図14に示したように、"DVR"ディレクトリを含むrootディレクトリ、"PLAYLIST"ディレクトリ、"CLIPINF"ディレクトリ、"M2TS"ディレクトリ、及び"DATA"ディレクトリを含む"DVR"ディレクトリである。rootディレクトリの下に、これら以外のディレクトリを作成されるようにしてもよいが、それらは、本例のアプリケーションフォーマットでは、無視されるとする。

"DVR"ディレクトリの下には、DVRアプリケーションフォーマットによって規定される全てのファイルとディレクトリがストアされる。"DVR"ディレクトリは、4個のディレクトリを含む。"PLAYLIST"ディレクトリの下には、Real PlayListとVirtual PlayListのデータベースファイルが置かれる。このディレクトリは、PlayListが1つもなくても存在する。

"CLIPINF"ディレクトリの下には、Clipのデータベースが置かれる。このディレクトリも、Clipが1つもなくても存在する。"M2TS"ディレクトリの下には、AVストリームファイルが置かれる。このディレクトリは、AVストリームファイルが1つもなくても存在する。"DATA"ディレクトリは、デジタルTV放送等のデータ放送のファイルがストアされる。

"DVR"ディレクトリは、次に示すファイルをストアする。"info.dvr"ファイルは、DVRディレクトリの下に作られ、アプリケーションレイヤの全体的な情報をストアする。DVRディレクトリの下には、ただ1つのinfo.dvrがなければならない。ファイル名は、info.dvrに固定されるとする。"menu.thmb"ファイルは、メニューサムネイル画像に関する情報をストアする。DVRディレクトリの下には、0又は1つのメニューサムネイルがなければならない。ファイル名は、menu.thmbに固定されるとする。メニューサムネイル画像が1つもない場合、このファイルは、存在しなくてもよい。

"mark.thmb"ファイルは、マークサムネイル画像に関する情報をストアする。DVRディレクトリの下には、0又は1つのマークサムネイルがなければならない。ファイル名は、mark.thmbに固定されるとする。メニューサムネイル画像が1つもない場合、このファイルは、存在しなくてもよい。

"PLAYLIST"ディレクトリは、2種類のPlayListファイルをストアするものであり、それらは、Real PlayListとVirtual PlayListである。"xxxxx.rpls" ファイ

ルは、1つのReal PlayListに関する情報をストアする。それぞれのReal PlayList毎に、1つのファイルが作られる。ファイル名は、"xxxxx.rpls"である。ここで、"xxxxx"は、5個の0乃至9までの数字である。ファイル拡張子は、"rpls"でなければならぬとする。

"yyyyy.vpls"ファイルは、1つのVirtual PlayListに関する情報をストアする。それぞれのVirtual PlayList毎に、1つのファイルが作られる。ファイル名は、"yyyyy.vpls"である。ここで、"yyyyy"は、5個の0乃至9までの数字である。ファイル拡張子は、"vpls"でなければならぬとする。

"CLIPINF"ディレクトリは、それぞれのAVストリームファイルに対応して、1つのファイルをストアする。"zzzzz.clpi"ファイルは、1つのAVストリームファイル(Clip AV stream file又はBridge-Clip AV stream file)に対応するClip Information fileである。ファイル名は、"zzzzz.clpi"であり、"zzzzz"は、5個の0乃至9までの数字である。ファイル拡張子は、"clpi"でなければならぬとする。

"M2TS"ディレクトリは、AVストリームのファイルをストアする。"zzzzz.m2ts"ファイルは、DVRシステムにより扱われるAVストリームファイルである。これは、Clip AV stream file又はBridge-Clip AV streamである。ファイル名は、"zzzzz.m2ts"であり、"zzzzz"は、5個の0乃至9までの数字である。ファイル拡張子は、"m2ts"でなければならぬとする。

"DATA"ディレクトリは、データ放送から伝送されるデータをストアするものであり、データとは、例えば、XML fileやMHEGファイル等である。

次に、各ディレクトリ(ファイル)のシンタクスとセマンティクスを説明する。先ず、"info.dvr"ファイルについて説明する。図15は、"info.dvr"ファイルのシンタクスを示す図である。"info.dvr"ファイルは、3個のオブジェクトから構成され、それらは、DVRVolume()、TableOfPlayLists()、及びMakerPrivateData()である。

図15に示したinfo.dvrのシンタクスについて説明すると、TableOfPlayLists\_Start\_addressは、info.dvrファイルの先頭のバイトからの相対バイト数を単位として、TableOfPlayList()の先頭アドレスを示す。相対バイト数は0からカウン

トされる。

`MarkerPrivateData_Start_address`は、`info.dvr`ファイルの先頭のバイトからの相対バイト数を単位として、`MarkerPrivateData()`の先頭アドレスを示す。相対バイト数は0からカウントされる。`padding_word` (パディングワード) は、`info.dvr`のシンタクスに従って挿入される。N1とN2は、0又は任意の正の整数である。それぞれのパディングワードは、任意の値をとるようにしてよい。

`DVRVolume()`は、ボリューム (ディスク) の内容を記述する情報をストアする。図16は、`DVRVolume()`のシンタクスを示す図である。図16に示した`DVR Volume()`のシンタクスを説明すると、`version_number`は、この`DVRVolume()`のバージョンナンバを示す4個のキャラクター文字を示す。`version_number`は、ISO 646に従って、"0045"と符号化される。

`length`は、この`length`フィールドの直後から`DVRVolume()`の最後までの`DVRVolume()`のバイト数を示す32ビットの符号なし整数で表される。

`ResumeVolume()`は、ボリュームの中で最後に再生したReal Playlist又はVirtual Playlistのファイル名を記憶している。但し、Real Playlist又はVirtual Playlistの再生をユーザが中断したときの再生位置は、`PlaylistMark()`において定義される`resume-mark`にストアされる。

図17は、`ResumeVolume()`のシンタクスを示す図である。図17に示した`ResumeVolume()`のシンタクスを説明すると、`valid_flag`は、この1ビットのフラグが1にセットされている場合、`resume_PlayList_name`フィールドが有効であることを示し、このフラグが0にセットされている場合、`resume_PlayList_name`フィールドが無効であることを示す。

`resume_PlayList_name`の10バイトのフィールドは、リジュームされるべきReal Playlist又はVirtual Playlistのファイル名を示す。

図16に示した`DVRVolume()`のシンタクスのなかの、`UIAppInfoVolume`は、ボリュームについてのユーザインターフェースアプリケーションのパラメータをストアする。図18は、`UIAppInfoVolume`のシンタクスを示す図であり、そのセマンティクスを説明すると、`character_set`の8ビットのフィールドは、`Volume_name`フィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方

法は、図19に示される値に対応する。

`name_length`の8ビットフィールドは、`Volume_name`フィールドの中に示されるボリューム名のバイト長を示す。`Volume_name`のフィールドは、ボリュームの名称を示す。このフィールドの中の左から`name_length`数のバイト数が、有効なキャラクター文字であり、それはボリュームの名称を示す。`Volume_name`フィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていてもよい。

`Volume_protect_flag`は、ボリュームの中のコンテンツを、ユーザに制限することなしに見せてよいかどうかを示すフラグである。このフラグが1にセットされている場合、ユーザが正しくPIN番号（パスワード）を入力できたときだけ、そのボリュームのコンテンツを、ユーザに見せること（再生されること）が許可される。このフラグが0にセットされている場合、ユーザがPIN番号を入力しなくても、そのボリュームのコンテンツを、ユーザに見せることが許可される。

最初に、ユーザが、ディスクをプレーヤへ挿入した時点において、もしこのフラグが0にセットされているか、又は、このフラグが1にセットされていてもユーザがPIN番号を正しく入力できたならば、記録再生装置1は、そのディスクの中のPlayListの一覧を表示させる。それぞれのPlayListの再生制限は、`volume_protect_flag`とは無関係であり、それはUIAppInfoPlayList()の中に定義される`playback_control_flag`によって示される。

PINは、4個の0乃至9までの数字で構成され、それぞれの数字は、ISO／IEC 646に従って符号化される。`ref_thumbnail_index`のフィールドは、ボリュームに付加されるサムネイル画像の情報を示す。`ref_thumbnail_index`フィールドが、0xFFFFでない値の場合、そのボリュームにはサムネイル画像が付加されており、そのサムネイル画像は、`menu.thum`ファイルの中にストアされている。その画像は、`menu.thum`ファイルの中で`ref_thumbnail_index`の値を用いて参照される。`ref_thumbnail_index`フィールドが、0xFFFFである場合、そのボリュームにはサムネイル画像が付加されていないことを示す。

次に、図15に示したinfo.dvrのシンタクス内のTableOfPlayLists()について説明する。TableOfPlayLists()は、PlayList(Real PlayListとVirtual PlayList)のファイル名をストアする。ボリュームに記録されている全てのPlayListファ

イルは、TableOfPlayList()の中に含まれる。TableOfPlayLists()は、ボリュームの中のPlayListのデフォルトの再生順序を示す。

図20は、TableOfPlayLists()のシンタクスを示す図であり、そのシンタクスについて説明すると、TableOfPlayListsのversion\_numberは、このTableOfPlayListsのバージョンナンバを示す4個のキャラクター文字を示す。version\_numberは、ISO 646に従って、"0045"と符号化されなければならない。

lengthは、このlengthフィールドの直後からTableOfPlayLists()の最後までのTableOfPlayLists()のバイト数を示す32ビットの符号なしの整数である。number\_of\_PlayListsの16ビットのフィールドは、Playlist\_file\_nameを含むfor-loopのループ回数を示す。この数字は、ボリュームに記録されているPlaylistの数に等しくなければならない。Playlist\_file\_nameの10バイトの数字は、Playlistのファイル名を示す。

図21は、TableOfPlayLists()のシンタクスを他の例の構成を示す図である。図21に示したシンタクスは、図20に示したシンタクスに、UIAppinfoPlaylist(後述)を含ませた構成とされている。このように、UIAppinfoPlaylistを含ませた構成とすることで、TableOfPlayListsを読み出すだけで、メニュー画面を作成することが可能となる。ここでは、図20に示したシンタクスを用いるとして以下の説明をする。

図15に示したinfo.dvrのシンタクス内のMakersPrivateDataについて説明する。MakersPrivateDataは、記録再生装置1のメーカーが、各社の特別なアプリケーションのために、MakersPrivateData()の中にメーカーのプライベートデータを挿入できるように設けられている。各メーカーのプライベートデータは、これを定義したメーカーを識別するために標準化されたmaker\_IDを持つ。MakersPrivateData()は、1つ以上のmaker\_IDを含んでもよい。

所定のメーカーが、プライベートデータを挿入したいときに、既に他のメーカーのプライベートデータがMakersPrivateData()に含まれていた場合、他のメーカーは、既にある古いプライベートデータを消去するのではなく、新しいプライベートデータをMakersPrivateData()の中に追加するようとする。このように、本実施例では、複数のメーカーのプライベートデータが、1つのMakersPrivateData()に含まれ

ることが可能であるようにする。

図22は、MakersPrivateDataのシンタクスを示す図である。図22に示したMakersPrivateDataのシンタクスについて説明すると、version\_numberは、このMakersPrivateData()のバージョンナンバを示す4個のキャラクター文字を示す。version\_numberは、ISO 646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からMakersPrivateData()の最後までのMakersPrivateData()のバイト数を示す32ビットの符号なし整数を示す。

mpd\_blocks\_start\_addressは、MakersPrivateData()の先頭のバイトからの相対バイト数を単位として、最初のmpd\_block()の先頭バイトアドレスを示す。相対バイト数は0からカウントされる。number\_of\_maker\_entriesは、MakersPrivateData()の中に含まれているメーカプライベートデータのエントリ数を与える16ビットの符号なし整数である。MakersPrivateData()の中に、同じmaker\_IDの値を持つメーカプライベートデータが2個以上存在してはならない。

mpd\_block\_sizeは、1024バイトを単位として、1つのmpd\_blockの大きさを与える16ビットの符号なし整数である。例えば、mpd\_block\_size=1ならば、それは1つのmpd\_blockの大きさが1024バイトであることを示す。number\_of\_mpd\_blocksは、MakersPrivateData()の中に含まれるmpd\_blockの数を与える16ビットの符号なし整数である。maker\_IDは、そのメーカプライベートデータを作成したDVRシステムの製造メーカを示す16ビットの符号なし整数である。maker\_IDに符号化される値は、このDVRフォーマットのライセンサによって指定される。

maker\_model\_codeは、そのメーカプライベートデータを作成したDVRシステムのモデルナンバコードを示す16ビットの符号なし整数である。maker\_model\_codeに符号化される値は、このフォーマットのライセンスを受けた製造メーカによって設定される。start\_mpd\_block\_numberは、そのメーカプライベートデータが開始されるmpd\_blockの番号を示す16ビットの符号なし整数である。メーカプライベートデータの先頭データは、mpd\_blockの先頭にアラインされなければならない。start\_mpd\_block\_numberは、mpd\_blockのfor-loopの中の変数jに対応する。

mpd\_lengthは、バイト単位でメーカプライベートデータの大きさを示す32ビ

ットの符号なし整数である。mpd\_blockは、メーカプライベートデータがストアされる領域である。MakersPrivateData()の中の全てのmpd\_blockは、同じサイズでなければならない。

次に、Real PlayList fileとVirtual PlayList fileについて、換言すれば、xxxxx.rplsとyyyyy.vplsについて説明する。図23は、xxxxx.rpls (Real PlayList)、又は、yyyyy.vpls (Virtual PlayList) のシンタクスを示す図である。xxxxx.rplsとyyyyy.vplsは、同一のシンタクス構成をもつ。xxxxx.rplsとyyyyy.vplsは、それぞれ、3個のオブジェクトから構成され、それらは、Playlist()、PlaylistMark()、及びMakerPrivateData()である。

PlaylistMark\_Start\_addressは、Playlistファイルの先頭のバイトからの相対バイト数を単位として、PlaylistMark()の先頭アドレスを示す。相対バイト数は0からカウントされる。

MakerPrivateData\_Start\_addressは、Playlistファイルの先頭のバイトからの相対バイト数を単位として、MakerPrivateData()の先頭アドレスを示す。相対バイト数は0からカウントされる。

padding\_word (パディングワード) は、Playlistファイルのシンタクスにしたがって挿入され、N1とN2は、0又は任意の正の整数である。それぞれのパディングワードは、任意の値をとるようにしてよい。

ここで、既に、簡便に説明したが、Playlistについて更に説明する。ディスク内にある全てのReal PlayListによって、Bridge-Clip (後述) を除く全てのClipの中の再生区間が参照されていなければならない。且つ、2つ以上のReal PlayListが、それらのPlayItemで示される再生区間を同一のClipの中でオーバーラップさせてはならない。

図24Aから図24Cを参照して更に説明すると、図24Aに示したように、全てのClipは、対応するReal PlayListが存在する。この規則は、図24Bに示したように、編集作業が行われた後においても守られる。したがって、全てのClipは、何れかのReal PlayListを参照することにより、必ず視聴することが可能である。

図24Cに示したように、Virtual PlayListの再生区間は、Real PlayListの再

生区間又はBridge-Clipの再生区間に含まれていなければならぬ。どのVirtual PlayListにも参照されないBridge-Clipがディスクの中に存在してはならない。

RealPlayListは、PlayItemのリストを含むが、SubPlayItemを含んではならない。Virtual PlayListは、PlayItemのリストを含み、PlayList()の中に示されるCPI\_typeがEP\_map typeであり、且つPlayList\_typeが0（ビデオとオーディオを含むPlayList）である場合、Virtual PlayListは、1つのSubPlayItemを含むことができる。PlayList()では、SubPlayItemはオーディオのアフレコの目的にだけに使用される、そして、1つのVirtual PlayListが持つSubPlayItemの数は、0又は1でなければならない。

次に、PlayListについて説明する。図25は、PlayListのシンタクスを示す図である。図25に示したPlayListのシンタクスを説明すると、version\_numberは、このPlayList()のバージョンナンバを示す4個のキャラクター文字である。version\_numberは、ISO 646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からPlayList()の最後までのPlayList()のバイト数を示す32ビットの符号なし整数である。PlayList\_typeは、このPlayListのタイプを示す8ビットのフィールドであり、その一例を図26に示す。

CPI\_typeは、1ビットのフラグであり、PlayItem()及びSubPlayItem()によって参照されるClipのCPI\_typeの値を示す。1つのPlayListによって参照される全てのClipは、それらのCPI()の中に定義されるCPI\_typeの値が同じでなければならない。number\_of\_PlayItemsは、PlayListの中にあるPlayItemの数を示す16ビットのフィールドである。

所定のPlayItem()に対応するPlayItem\_idは、PlayItem()を含むfor-loopの中で、そのPlayItem()の現れる順番により定義される。PlayItem\_idは、0から開始される。number\_of\_SubPlayItemsは、PlayListの中にあるSubPlayItemの数を示す16ビットのフィールドである。この値は、0又は1である。付加的なオーディオストリームのバス（オーディオストリームバス）は、サブバスの一種である。

次に、図25に示したPlayListのシンタクスのUIAppInfoPlayListについて説明する。UIAppInfoPlayListは、PlayListについてのユーザインタフェースアプリケ

ーションのパラメータをストアする。図27は、UIAppInfoPlayListのシンタクスを示す図である。図27に示したUIAppInfoPlayListのシンタクスを説明するに、character\_setは、8ビットのフィールドであり、PlayList\_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示したテーブルに準拠する値に対応する。

name\_lengthは、8ビットフィールドであり、PlayList\_nameフィールドの中に示されるPlayList名のバイト長を示す。PlayList\_nameのフィールドは、PlayListの名称を示す。このフィールドの中の左からname\_length数のバイト数が、有効なキャラクター文字であり、それはPlayListの名称を示す。PlayList\_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていてもよい。

record\_time\_and\_dateは、PlayListが記録されたときの日時をストアする56ビットのフィールドである。このフィールドは、年／月／日／時／分／秒について、14個の数字を4ビットのBinary Coded Decimal (BCD) で符号化したものである。例えば、2001/12/23:01:02:03は、"0x20011223010203"と符号化される。

durationは、PlayListの総再生時間を時間／分／秒の単位で示した24ビットのフィールドである。このフィールドは、6個の数字を4ビットのBinary Coded Decimal (BCD) で符号化したものである。例えば、01:45:30は、"0x014530"と符号化される。

valid\_periodは、PlayListが有効である期間を示す32ビットのフィールドである。このフィールドは、8個の数字を4ビットのBinary Coded Decimal (BCD) で符号化したものである。例えば、記録再生装置1は、この有効期間の過ぎたPlayListを自動消去する、といったように用いられる。例えば、2001/05/07は、"0x20010507"と符号化される。

maker\_idは、そのPlayListを最後に更新したDVRプレーヤ（記録再生装置1）の製造者を示す16ビットの符号なし整数である。maker\_idに符号化される値は、DVRフォーマットのライセンサによって割り当てられる。maker\_codeは、そのPlayListを最後に更新したDVRプレーヤのモデル番号を示す16ビットの

符号なし整数である。`maker_code`に符号化される値は、DVRフォーマットのライセンスを受けた製造者によって決められる。

`playback_control_flag`のフラグが1にセットされている場合、ユーザが正しくPIN番号を入力できた場合にだけ、そのPlayListは再生される。このフラグが0にセットされている場合、ユーザがPIN番号を入力しなくても、ユーザは、そのPlayListを視聴することができる。

`write_protect_flag`は、図28Aにテーブルを示すように、1にセットされている場合、`write_protect_flag`を除いて、そのPlayListの内容は、消去及び変更されない。このフラグが0にセットされている場合、ユーザは、そのPlayListを自由に消去及び変更できる。このフラグが1にセットされている場合、ユーザが、そのPlayListを消去、編集、又は上書きする前に、記録再生装置1はユーザに再確認するようなメッセージを表示させる。

`write_protect_flag`が0にセットされているReal PlayListが存在し、且つ、そのReal PlayListのClipを参照するVirtual PlayListが存在し、そのVirtual PlayListの`write_protect_flag`が1にセットされていてもよい。ユーザが、Real PlayListを消去しようとする場合、記録再生装置1は、そのReal PlayListを消去する前に、上記Virtual PlayListの存在をユーザに警告するか、又は、そのReal PlayListを"Minimize"する。

`is_played_flag`は、図28Bに示すように、フラグが1にセットされている場合、そのPlayListは、記録されてから一度は再生されたことを示し、0にセットされている場合、そのPlayListは、記録されてから一度も再生されたことがないことを示す。

`archive`は、図28Cに示すように、そのPlayListがオリジナルであるか、コピーされたものであるかを示す2ビットのフィールドである。`ref_thumbnail_index`のフィールドは、PlayListを代表するサムネイル画像の情報を示す。`ref_thumbnail_index`フィールドが、0xFFFFでない値の場合、そのPlayListには、PlayListを代表するサムネイル画像が付加されており、そのサムネイル画像は、`menu.thumb`ファイルの中にストアされている。その画像は、`menu.thumb`ファイルの中で`ref_thumbnail_index`の値を用いて参照される。`ref_thumbnail_index`フィールドが、

0xFFFF である場合、そのPlayListには、PlayListを代表するサムネイル画像が付加されていない。

次に、PlayItemについて説明する。1つのPlayItem()は、基本的に次のデータを含む。Clipのファイル名を指定するためのClip\_information\_file\_name、Clipの再生区間を特定するためのIN\_timeとOUT\_timeのペア、PlayList()において定義されるCPI\_typeがEP\_map typeである場合、IN\_timeとOUT\_timeが参照するところのSTC\_sequence\_id、及び、先行するPlayItemと現在のPlayItemとの接続の状態を示すところのconnection\_conditionである。

PlayListが2つ以上のPlayItemから構成されるとき、これらのPlayItemはPlayListのグローバル時間軸上に、時間のギャップ又はオーバーラップなしに一列に並べられる。PlayList()において定義されるCPI\_typeがEP\_map typeであり、且つ現在のPlayItemがBridgeSequence()を持たないとき、そのPlayItemにおいて定義されるIN\_timeとOUT\_timeのペアは、STC\_sequence\_idによって指定される同じS T C連続区間上の時間を指していなければならない。このような例を図29に示す。

図30は、PlayList()において定義されるCPI\_typeがEP\_map typeであり、且つ現在のPlayItemがBridgeSequence()を持つとき、次に説明する規則が適用される場合を示している。現在のPlayItemに先行するPlayItemのIN\_time（図の中でIN\_time1と示されているもの）は、先行するPlayItemのSTC\_sequence\_idによって指定されるS T C連続区間上の時間を指している。先行するPlayItemのOUT\_time（図の中でOUT\_time1と示されているもの）は、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このOUT\_timeは、後述する符号化制限に従っていなければならない。

現在のPlayItemのIN\_time（図の中でIN\_time2と示されているもの）は、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このIN\_timeも、後述する符号化制限に従っていなければならない。現在のPlayItemのPlayItemのOUT\_time（図の中でOUT\_time2と示されているもの）は、現在のPlayItemのSTC\_sequence\_idによって指定されるS T C連続区間上の時間を指している。

図3 1に示すように、PlayList()のCPI\_typeがTU\_map typeである場合、PlayItemのIN\_timeとOUT\_timeのペアは、同じClip AVストリーム上の時間を指している。

PlayItemのシンタクスは、図3 2に示すようになる。図3 2に示したPlayItemのシンタクスを説明すると、Clip\_Information\_file\_nameのフィールドは、Clip Information fileのファイル名を示す。このClip Information fileのClipInfo()において定義されるClip\_stream\_typeは、Clip AV streamを示していなければならない。

STC\_sequence\_idは、8ビットのフィールドであり、PlayItemが参照するSTC連続区間のSTC\_sequence\_idを示す。PlayList()の中で指定されるCPI\_typeがTU\_map typeである場合、この8ビットフィールドは何も意味を持たず、0にセットされる。IN\_timeは、32ビットフィールドであり、PlayItemの再生開始時刻をストアする。IN\_timeのセマンティクスは、図3 3に示すように、PlayList()において定義されるCPI\_typeによって異なる。

OUT\_timeは、32ビットフィールドであり、PlayItemの再生終了時刻をストアする。OUT\_timeのセマンティクスは、図3 4に示すように、PlayList()において定義されるCPI\_typeによって異なる。

Connection\_Conditionは、図3 5に示したような先行するPlayItemと、現在のPlayItemとの間の接続状態を示す2ビットのフィールドである。図3 6 A～図3 6 Dは、図3 5に示したConnection\_Conditionの各状態について説明する図である。

次に、BridgeSequenceInfoについて、図3 7を参照して説明する。BridgeSequenceInfo()は、現在のPlayItemの付属情報であり、次に示す情報を持つ。Bridge-Clip AV streamファイルとそれに対応するClip Information fileを指定するBridge\_Clip\_Information\_file\_nameを含む。

また、先行するPlayItemが参照するClip AV stream上のソースパケットのアドレスであり、このソースパケットに統いてBridge-Clip AV streamファイルの最初のソースパケットが接続される。このアドレスは、RSPN\_exit\_from\_previous\_Clipと称される。更に現在のPlayItemが参照するClip AV stream上のソースパケッ

トのアドレスであり、このソースパケットの前にBridge-Clip AV streamファイルの最後のソースパケットが接続される。このアドレスは、RSPN\_enter\_to\_current\_Clipという。

図37において、RSPN\_arrival\_time\_discontinuityは、the Bridge-Clip AV streamファイルの中でアライバルタイムベースの不連続点があるところのソースパケットのアドレスを示す。このアドレスは、ClipInfo()の中において定義される。

図38は、BridgeSequenceinfoのシンタクスを示す図である。図38に示したBridgeSequenceinfoのシンタクスを説明すると、Bridge\_Clip\_Information\_file\_nameのフィールドは、Bridge-Clip AV streamファイルに対応するClip Information fileのファイル名を示す。このClip Information fileのClipInfo()において定義されるClip\_stream\_typeは、'Bridge-Clip AV stream'を示していなければならない。

RSPN\_exit\_from\_previous\_Clipの32ビットフィールドは、先行するPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、このソースパケットに続いてBridge-Clip AV streamファイルの最初のソースパケットが接続される。RSPN\_exit\_from\_previous\_Clipは、ソースパケット番号を単位とする大きさであり、先行するPlayItemが参照するClip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset\_SPNの値を初期値としてカウントされる。

RSPN\_enter\_to\_current\_Clipの32ビットフィールドは、現在のPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、このソースパケットの前にBridge-Clip AV streamファイルの最後のソースパケットが接続される。RSPN\_exit\_from\_previous\_Clipは、ソースパケット番号を単位とする大きさであり、現在のPlayItemが参照するClip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset\_SPNの値を初期値としてカウントされる。

次に、SubPlayItemについて、図39を参照して説明する。SubPlayItem()の使用は、PlayList()のCPI\_typeがEP\_map\_typeである場合だけに許される。ここでは、

SubPlayItemはオーディオのアフレコの目的のためだけに使用されるとする。SubPlayItem()は、次に示すデータを含む。先ず、PlayListの中のsub pathが参照するClipを指定するためのClip\_information\_file\_nameを含む。

また、Clipの中のsub pathの再生区間を指定するためのSubPath\_IN\_time と SubPath\_OUT\_timeを含む。更に、main pathの時間軸上でsub pathが再生開始する時刻を指定するためのsync\_PlayItem\_id と sync\_start PTS\_of\_PlayItemを含む。sub pathに参照されるオーディオのClip AV streamは、STC不連続点（システムタイムベースの不連続点）を含んではならない。sub pathに使われるClipのオーディオサンプルのクロックは、main pathのオーディオサンプルのクロックにロックされている。

図40は、SubPlayItemのシンタクスを示す図である。図40に示したSubPlayItemのシンタクスを説明すると、Clip\_Information\_file\_nameのフィールドは、Clip Information fileのファイル名を示し、それはPlayListの中でsub pathによって使用される。このClip Information fileのClipInfo()において定義されるClip\_stream\_typeは、Clip AV streamを示していかなければならない。

SubPath\_typeの8ビットのフィールドは、sub pathのタイプを示す。ここでは、図41に示すように、'0x00'しか設定されておらず、他の値は、将来のために確保されている。

sync\_PlayItem\_idの8ビットのフィールドは、main pathの時間軸上でsub pathが再生開始する時刻が含まれるPlayItemのPlayItem\_idを示す。所定のPlayItemに対応するPlayItem\_idの値は、PlayList()において定義される（図25参照）。

sync\_start PTS\_of\_PlayItemの32ビットのフィールドは、main pathの時間軸上でsub pathが再生開始する時刻を示し、sync\_PlayItem\_idで参照されるPlayItem上のPTS (Presentation Time Stamp)の上位32ビットを示す。SubPath\_IN\_timeの32ビットフィールドは、Sub pathの再生開始時刻をストアする。SubPath\_IN\_timeは、Sub Pathの中で最初のプレゼンテーションユニットに対応する33ビット長のPTSの上位32ビットを示す。

SubPath\_OUT\_timeの32ビットフィールドは、Sub pathの再生終了時刻をストアする。SubPath\_OUT\_timeは、次式によって算出されるPresentation\_end\_TSの値

の上位 32 ビットを示す。

`Presentation_end_TS = PTS_out + AU_duration`

ここで、`PTS_out`は、`SubPath`の最後のプレゼンテーションユニットに対応する33ビット長の PTS である。`AU_duration`は、`SubPath`の最後のプレゼンテーションユニットの 90 kHz 単位の表示期間である。

次に、図 23 に示した `xxxxx.rpls` と `yyyyy.vpls` のシンタクス内の `PlayListMark()` について説明する。PlayListについてのマーク情報は、この `PlayListMark` にストアされる。図 42 は、`PlayListMark` のシンタクスを示す図である。図 42 に示した `PlayListMark` のシンタクスについて説明すると、`version_number` は、この `PlayListMark()` のバージョンナンバを示す 4 個のキャラクター文字である。`version_number` は、ISO 646 に従って、"0045" と符号化されなければならない。

`length` は、この `length` フィールドの直後から `PlayListMark()` の最後までの `PlayListMark()` のバイト数を示す 32 ビットの符号なし整数である。`number_of_PlayList_marks` は、`PlayListMark` の中にストアされているマークの個数を示す 16 ビットの符号なし整数である。`number_of_PlayList_marks` は、0 であってもよい。`mark_type` は、マークのタイプを示す 8 ビットのフィールドであり、図 43 に示すテーブルに従って符号化される。

`mark_time_stamp` の 32 ビットフィールドは、マークが指定されたポイントを示すタイムスタンプをストアする。`mark_time_stamp` のセマンティクスは、図 44 に示すように、`PlayList()` において定義される `CPI_type` によって異なる。`PlayItem_id` は、マークが置かれているところの `PlayItem` を指定する 8 ビットのフィールドである。所定の `PlayItem` に対応する `PlayItem_id` の値は、`PlayList()` において定義される（図 25 参照）。

`character_set` の 8 ビットのフィールドは、`mark_name` フィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図 19 に示した値に対応する。`name_length` の 8 ビットフィールドは、`Mark_name` フィールドの中に示されるマーク名のバイト長を示す。`mark_name` のフィールドは、マークの名称を示す。このフィールドの中の左から `name_length` 数のバイト数が、有効なキャラクター文字であり、それはマークの名称を示す。`Mark_name` フィールドの中で、

それら有効なキャラクター文字の後の値は、どのような値が設定されてもよい。

`ref_thumbnail_index`のフィールドは、マークに付加されるサムネイル画像の情報表示。`ref_thumbnail_index`フィールドが、`0xFFFF`でない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、`mark.thmb`ファイルの中にストアされている。その画像は、`mark.thmb`ファイルの中で`ref_thumbnail_index`の値を用いて参照される（後述）。`ref_thumbnail_index`フィールドが、`0xFFFF`である場合、そのマークにはサムネイル画像が付加されていないことを示す。

次に、`Clip information file`について説明する。`zzzzz.clpi` (`Clip information file`ファイル) は、図45に示すように6個のオブジェクトから構成される。これらは、`ClipInfo()`、`STC_Info()`、`ProgramInfo()`、`CPI()`、`ClipMark()`、及び`MakerPrivateData()`である。AVストリーム(Clip AVストリーム又はBridge-Clip AV stream)とそれに対応するClip Informationファイルは、同じ数字列の”`zzzz`”が使用される。

図45に示した`zzzzz.clpi` (`Clip information file`ファイル) のシンタクスについて説明すると、`ClipInfo_Start_address`は、`zzzzz.clpi`ファイルの先頭のバイトからの相対バイト数を単位として、`ClipInfo()`の先頭アドレスを示す。相対バイト数は0からカウントされる。

`STC_Info_Start_address`は、`zzzzz.clpi`ファイルの先頭のバイトからの相対バイト数を単位として、`STC_Info()`の先頭アドレスを示す。相対バイト数は0からカウントされる。`ProgramInfo_Start_address`は、`zzzzz.clpi`ファイルの先頭のバイトからの相対バイト数を単位として、`ProgramInfo()`の先頭アドレスを示す。相対バイト数は0からカウントされる。`CPI_Start_address`は、`zzzzz.clpi`ファイルの先頭のバイトからの相対バイト数を単位として、`CPI()`の先頭アドレスを示す。相対バイト数は0からカウントされる。

`ClipMark_Start_address`は、`zzzzz.clpi`ファイルの先頭のバイトからの相対バイト数を単位として、`ClipMark()`の先頭アドレスを示す。相対バイト数は0からカウントされる。`MakerPrivateData_Start_address`は、`zzzzz.clpi`ファイルの先頭のバイトからの相対バイト数を単位として、`MakerPrivateData()`の先頭アドレスを示す。

スを示す。相対バイト数は 0 からカウントされる。padding\_word (パディングワード) は、zzzz.clpiファイルのシンタクスにしたがって挿入される。N1, N2, N3, N4、及びN5は、0 又は任意の正の整数でなければならない。それぞれのパディングワードは、任意の値がとられるようにしてもよい。

次に、ClipInfoについて説明する。図46は、ClipInfoのシンタクスを示す図である。ClipInfo()は、それに対応するAVストリームファイル (Clip AVストリーム又はBridge-Clip AVストリームファイル) の属性情報をストアする。

図46に示したClipInfoのシンタクスについて説明すると、version\_numberは、このClipInfo()のバージョンナンバを示す4個のキャラクター文字である。version\_numberは、ISO 646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からClipInfo()の最後までのClipInfo()のバイト数を示す32ビットの符号なし整数である。Clip\_stream\_typeの8ビットのフィールドは、図47に示すように、Clip Informationファイルに対応するAVストリームのタイプを示す。それぞれのタイプのAVストリームのストリームタイプについては後述する。

offset\_SPNの32ビットのフィールドは、AVストリーム (Clip AVストリーム又はBridge-Clip AVストリーム) ファイルの最初のソースパケットについてのソースパケット番号のオフセット値を与える。AVストリームファイルが最初にディスクに記録されるとき、このoffset\_SPNは0でなければならない。

図48に示すように、AVストリームファイルのはじめの部分が編集によって消去されたとき、offset\_SPNは、0以外の値をとってもよい。ここでは、offset\_SPNを参照する相対ソースパケット番号 (相対アドレス) が、しばしば、RSPN\_xx (xxは変形する。例。RSPN\_EP\_start) の形式でシンタクスの中に記述されている。相対ソースパケット番号は、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからoffset\_SPNの値を初期値としてカウントされる。

AVストリームファイルの最初のソースパケットから相対ソースパケット番号で参照されるソースパケットまでのソースパケットの数 (SPN\_xxx) は、次式で算出される。

```
SPN_xxx = RSPN_xxx - offset_SPN
```

図48に、offset\_SPNが4である場合の例を示す。

TS\_recording\_rateは、24ビットの符号なし整数であり、この値は、DVRドライブ（書き部22）へ又はDVRドライブ（読み出部28）からのAVストリームの必要な入出力のピットレートを与える。record\_time\_and\_dateは、Clipに対応するAVストリームが記録されたときの日時をストアする56ビットのフィールドであり、年/月/日/時/分/秒について、14個の数字を4ビットのBinary Coded Decimal (BCD)で符号化したものである。例えば、2001/12/23:01:02:03は、"0x20011223010203"と符号化される。

durationは、Clipの総再生時間をアライバルタイムクロックに基づいた時間/分/秒の単位で示した24ビットのフィールドである。このフィールドは、6個の数字を4ビットのBinary Coded Decimal (BCD)で符号化したものである。例えば、01:45:30は、"0x014530"と符号化される。

time\_controlled\_flag:のフラグは、AVストリームファイルの記録モードを示す。このtime\_controlled\_flagが1である場合、記録モードは、記録してからの時間経過に対してファイルサイズが比例するようにして記録されるモードであることを示し、次式に示す条件を満たさなければならない。

$$\begin{aligned} & TS_{\text{average\_rate}} * 192 / 188 * (t - \text{start\_time}) - \alpha \leq \text{size\_clip}(t) \\ & \leq TS_{\text{average\_rate}} * 192 / 188 * (t - \text{start\_time}) + \alpha \end{aligned}$$

ここで、TS\_average\_rateは、AVストリームファイルのトランスポートストリームの平均ピットレートをbytes/secondの単位で表したものである。

また、上式において、tは、秒単位で表される時間を示し、start\_timeは、AVストリームファイルの最初のソースパケットが記録されたときの時刻であり、秒単位で表される。size\_clip(t)は、時刻tにおけるAVストリームファイルのサイズをバイト単位で表したものであり、例えば、start\_timeから時刻tまでに10個のソースパケットが記録された場合、size\_clip(t)は10\*192バイトである。 $\alpha$ は、TS\_average\_rateに依存する定数である。

time\_controlled\_flagが0にセットされている場合、記録モードは、記録の時間経過とAVストリームのファイルサイズが比例するように制御していないことを

示す。例えば、これは入力トランSPORTストリームをトランスペアレント記録する場合である。

TS\_average\_rateは、time\_controlled\_flagが1にセットされている場合、この24ビットのフィールドは、上式で用いているTS\_average\_rateの値を示す。time\_controlled\_flagが0にセットされている場合、このフィールドは、何も意味を持たず、0にセットされなければならない。例えば、可変ビットレートのトランSPORTストリームは、次に示す手順により符号化される。先ずトランSPORTレートをTS\_recording\_rateの値にセットする。次に、ビデオストリームを可変ビットレートで符号化する。そして、ヌルパケットを使用しないことによって、間欠的にトランSPORTパケットを符号化する。

RSPN\_arrival\_time\_discontinuityの32ビットフィールドは、Bridge-Clip AV streamファイル上でアライバルタイムベースの不連続が発生する場所の相対アドレスである。RSPN\_arrival\_time\_discontinuityは、ソースパケット番号を単位とする大きさであり、Bridge-Clip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset\_SPNの値を初期値としてカウントされる。そのBridge-Clip AV streamファイルの中での絶対アドレスは、上述した

SPN\_xxx = RSPN\_xxx - offset\_SPN

に基づいて算出される。

reserved\_for\_system\_useの144ビットのフィールドは、システム用にリザーブされている。is\_format\_identifier\_validのフラグが1であるとき、format\_identifierのフィールドが有効であることを示す。is\_original\_network\_ID\_validのフラグが1である場合、original\_network\_IDのフィールドが有効であることを示す。is\_transport\_stream\_ID\_validのフラグが1である場合、transport\_stream\_IDのフィールドが有効であることを示す。is\_servece\_ID\_validのフラグが1である場合、servece\_IDのフィールドが有効であることを示す。

is\_country\_code\_validのフラグが1であるとき、country\_codeのフィールドが有効であることを示す。format\_identifierの32ビットフィールドは、トランSPORTストリームの中でregistration deascriotor (ISO/IEC13818-1で定義されている) が持つformat\_identifierの値を示す。original\_network\_IDの16ビ

ットフィールドは、トランSPORTストリームの中で定義されているoriginal\_network\_IDの値を示す。transport\_stream\_IDの16ビットフィールドは、トランSPORTストリームの中で定義されているtransport\_stream\_IDの値を示す。

servece\_IDの16ビットフィールドは、トランSPORTストリームの中で定義されているservece\_IDの値を示す。country\_codeの24ビットのフィールドは、ISO3166によって定義されるカントリーコードを示す。それぞれのキャラクター文字は、ISO8859-1で符号化される。例えば、日本は”JPN”と表され、”0x4A 0x50 0x4E”と符号化される。stream\_format\_nameは、トランSPORTストリームのストリーム定義をしているフォーマット機関の名称を示すISO-646の16個のキャラクターコードである。このフィールドの中の無効なバイトは、値’0xFF’がセットされる。

format\_identifier、original\_network\_ID、transport\_stream\_ID、servece\_ID、country\_code、及びstream\_format\_nameは、トランSPORTストリームのサービスプロバイダを示すものであり、これにより、オーディオやビデオストリームの符号化制限、SI(サービスインフォメーション)の規格やオーディオビデオストリーム以外のプライベートデータストリームのストリーム定義を認識することができる。これらの情報は、デコーダが、そのストリームをデコードできるか否か、そしてデコードできる場合にデコード開始前にデコーダシステムの初期設定を行うために用いることが可能である。

次に、STC\_Infoについて説明する。ここでは、MPEG-2トランSPORTストリームの中でSTCの不連続点(システムタイムベースの不連続点)を含まない時間区間をSTC\_sequenceと称し、Clipの中で、STC\_sequenceは、STC\_sequence\_idの値によって特定される。図50A及び図50Bは、連続なSTC区間について説明する図である。同じSTC\_sequenceの中で同じSTCの値は、決して現れない(但し、後述するように、Clipの最大時間長は制限されている)。したがって、同じSTC\_sequenceの中で同じPTSの値もまた、決して現れない。AVストリームが、N(N>0)個のSTC不連続点を含む場合、Clipのシステムタイムベースは、(N+1)個のSTC\_sequenceに分割される。

STC\_Infoは、STCの不連続(システムタイムベースの不連続)が発生する場

所のアドレスをストアする。図 5 1 を参照して説明するように、RSPN\_STC\_start が、そのアドレスを示し、最後のSTC\_sequenceを除くk番目 ( $k >= 0$ ) のSTC\_sequenceは、k番目のRSPN\_STC\_startで参照されるソースパケットが到着した時刻から始まり、(k + 1) 番目のRSPN\_STC\_startで参照されるソースパケットが到着した時刻で終わる。最後のSTC\_sequenceは、最後のRSPN\_STC\_startで参照されるソースパケットが到着した時刻から始まり、最後のソースパケットが到着した時刻で終了する。

図 5 2 は、STC\_Infoのシンタクスを示す図である。図 5 2 に示したSTC\_Infoのシンタクスについて説明すると、version\_numberは、このSTC\_Info()のバージョンナンバを示す4個のキャラクター文字である。version\_numberは、ISO 646に従って、"0045"と符号化されなければならない。

lengthは、このlengthフィールドの直後からSTC\_Info()の最後までのSTC\_Info()のバイト数を示す32ビットの符号なし整数である。CPI()のCPI\_typeがTU\_map\_typeを示す場合、このlengthフィールドは0をセットしてもよい。CPI()のCPI\_typeがEP\_map\_typeを示す場合、num\_of\_STC\_sequencesは1以上の値でなければならない。

num\_of\_STC\_sequencesの8ビットの符号なし整数は、Clipの中でのSTC\_sequenceの数を示す。この値は、このフィールドに続くfor-loopのループ回数を示す。所定のSTC\_sequenceに対応するSTC\_sequence\_idは、RSPN\_STC\_startを含むfor-loopの中で、そのSTC\_sequenceに対応するRSPN\_STC\_startの現れる順番により定義されるものである。STC\_sequence\_idは、0から開始される。

RSPN\_STC\_startの32ビットフィールドは、AVストリームファイル上でSTC\_sequenceが開始するアドレスを示す。RSPN\_STC\_startは、AVストリームファイルの中でシステムタイムベースの不連続点が発生するアドレスを示す。RSPN\_STC\_startは、AVストリームの中で新しいシステムタイムベースの最初のPCRを持つソースパケットの相対アドレスとしてもよい。RSPN\_STC\_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset\_SPNの値を初期値としてカウントされる。そのAV streamファイルの中での絶対アドレスは、既に上述した

`SPN_xxx = RSPN_xxx - offset_SPN`  
により算出される。

次に、図45に示したzzzz.clipのシンタクス内のProgramInfoについて説明する。図53を参照しながら説明すると、ここでは、Clipの中で次の特徴をもつ時間区間をprogram\_sequenceと呼ぶ。先ず、PCR\_PIDの値が変わらない。次に、ビデオエレメンタリーストリームの数が変化しない。また、それぞれのビデオストリームについてのPIDの値とそのVideoCodingInfoによって定義される符号化情報が変化しない。更に、オーディオエレメンタリーストリームの数が変化しない。また、それぞれのオーディオストリームについてのPIDの値とそのAudioCodingInfoによって定義される符号化情報が変化しない。

program\_sequenceは、同一の時刻において、ただ1つのシステムタイムベースを持つ。program\_sequenceは、同一の時刻において、ただ1つのPMTを持つ。ProgramInfo()は、program\_sequenceが開始する場所のアドレスをストアする。RSPN\_program\_sequence\_startが、そのアドレスを示す。

図54は、ProgramInfoのシンタクスを示す図である。図54に示したProgramInfoのシンタクを説明すると、version\_numberは、このProgramInfo()のバージョンナンバを示す4個のキャラクター文字である。version\_numberは、ISO 646に従って、"0045"と符号化されなければならない。

lengthは、このlengthフィールドの直後からProgramInfo()の最後までのProgramInfo()のバイト数を示す32ビットの符号なし整数である。CPI()のCPI\_typeがTU\_map\_typeを示す場合、このlengthフィールドは0にセットされてもよい。CPI()のCPI\_typeがEP\_map\_typeを示す場合、number\_of\_programsは1以上の値でなければならない。

number\_of\_program\_sequencesの8ビットの符号なし整数は、Clipの中でのprogram\_sequenceの数を示す。この値は、このフィールドに続くfor-loopのループ回数を示す。Clipの中でprogram\_sequenceが変化しない場合、number\_of\_program\_sequencesは1をセットされなければならない。RSPN\_program\_sequence\_startの32ビットフィールドは、AVストリームファイル上でプログラムシーケンスが開始する場所の相対アドレスである。

RSPN\_program\_sequence\_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset\_SPNの値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、

$$\text{SPN\_xxx} = \text{RSPN\_xxx} - \text{offset\_SPN}$$

により算出される。シンタクスのfor-loopの中でRSPN\_program\_sequence\_start値は、昇順に現れなければならない。

PCR\_PIDの16ビットフィールドは、そのprogram\_sequenceに有効なPCRフィールドを含むトランスポートパケットのPIDを示す。number\_of\_videosの8ビットフィールドは、video\_stream\_PIDとVideoCodingInfo()を含むfor-loopのループ回数を示す。number\_of\_audiosの8ビットフィールドは、audio\_stream\_PIDとAudioCodingInfo()を含むfor-loopのループ回数を示す。video\_stream\_PIDの16ビットフィールドは、そのprogram\_sequenceに有効なビデオストリームを含むトランスポートパケットのPIDを示す。このフィールドに続くVideoCodingInfo()は、そのvideo\_stream\_PIDで参照されるビデオストリームの内容を説明しなければならない。

audio\_stream\_PIDの16ビットフィールドは、そのprogram\_sequenceに有効なオーディオストリームを含むトランスポートパケットのPIDを示す。このフィールドに続くAudioCodingInfo()は、そのaudio\_stream\_PIDで参照されるビデオストリームの内容を説明しなければならない。

なお、シンタクスのfor-loopの中でvideo\_stream\_PIDの値の現れる順番は、そのprogram\_sequenceに有効なPMTの中でビデオストリームのPIDが符号化されている順番に等しくなければならない。また、シンタクスのfor-loopの中でaudio\_stream\_PIDの値の現れる順番は、そのprogram\_sequenceに有効なPMTの中でオーディオストリームのPIDが符号化されている順番に等しくなければならない。

図55は、図54に示したPrograminfoのシンタクス内のVideoCodingInfoのシンタクスを示す図である。図55に示したVideoCodingInfoのシンタクスを説明すると、video\_formatの8ビットフィールドは、図56に示すように、ProgramInf

o()の中のvideo\_stream\_PIDに対応するビデオフォーマットを示す。

frame\_rateの8ビットフィールドは、図57に示すように、ProgramInfo()の中のvideo\_stream\_PIDに対応するビデオのフレームレートを示す。display\_aspect\_ratioの8ビットフィールドは、図58に示すように、ProgramInfo()の中のvideo\_stream\_PIDに対応するビデオの表示アスペクト比を示す。

図59は、図54に示したProgramInfoのシンタクス内のAudioCodingInfoのシンタクスを示す図である。図59に示したAudioCodingInfoのシンタクスを説明すると、audio\_codingの8ビットフィールドは、図60に示すように、ProgramInfo()の中のaudio\_stream\_PIDに対応するオーディオの符号化方法を示す。

audio\_component\_typeの8ビットフィールドは、図61に示すように、ProgramInfo()の中のaudio\_stream\_PIDに対応するオーディオのコンポーネントタイプを示す。sampling\_frequencyの8ビットフィールドは、図62に示すように、ProgramInfo()の中のaudio\_stream\_PIDに対応するオーディオのサンプリング周波数を示す。

次に、図45に示したzzzzz.clipのシンタクス内のCPI (Characteristic Point Information)について説明する。CPIは、AVストリームの中の時間情報とそのファイルの中のアドレスとを関連づけるためにある。CPIには2つのタイプがあり、これらはEP\_mapとTU\_mapである。図63に示すように、CPI()の中のCPI\_typeがEP\_map typeの場合、そのCPI()はEP\_mapを含む。図64に示すように、CPI()の中のCPI\_typeがTU\_map typeの場合、そのCPI()はTU\_mapを含む。1つのAVストリームは、1つのEP\_map又は1つのTU\_mapを持つ。AVストリームがSESFトランスポートストリームの場合、それに対応するClipはEP\_mapを持たなければならない。

図65は、CPIのシンタクスを示す図である。図65に示したCPIのシンタクスを説明すると、version\_numberは、このCPI()のバージョンナンバを示す4個のキャラクター文字である。version\_numberは、ISO 646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からCPI()の最後までのCPI()のバイト数を示す32ビットの符号なし整数である。CPI\_typeは、図66に示すように、1ビットのフラグであり、ClipのCPIのタイ

ブを表す。

次に、図65に示したCPIのシンタクス内のEP\_mapについて説明する。EP\_mapには、2つのタイプがあり、それはビデオストリーム用のEP\_mapとオーディオストリーム用のEP\_mapである。EP\_mapの中のEP\_map\_typeが、EP\_mapのタイプを区別する。Clipが1つ以上のビデオストリームを含む場合、ビデオストリーム用のEP\_mapが使用されなければならない。Clipがビデオストリームを含まず、1つ以上のオーディオストリームを含む場合、オーディオストリーム用のEP\_mapが使用されなければならない。

ビデオストリーム用のEP\_mapについて図67を参照して説明する。ビデオストリーム用のEP\_mapは、stream\_PID、PTS\_EP\_start、及び、RSPN\_EP\_startというデータを持つ。stream\_PIDは、ビデオストリームを伝送するトランスポートパケットのP I Dを示す。PTS\_EP\_startは、ビデオストリームのシーケンスヘッダから始めるアクセスユニットのP T Sを示す。RSPN\_EP\_startは、AVストリームの中でPTS\_EP\_startにより参照されるアクセスユニットの第1バイト目を含むソースポケットのアドレスを示す。

EP\_map\_for\_one\_stream\_PID()と呼ばれるサブテーブルは、同じP I Dを持つトランスポートパケットによって伝送されるビデオストリーム毎に作られる。Clipの中に複数のビデオストリームが存在する場合、EP\_mapは複数のEP\_map\_for\_one\_stream\_PID()を含んでもよい。

オーディオストリーム用のEP\_mapは、stream\_PID、PTS\_EP\_start、及びRSPN\_EP\_startというデータを持つ。stream\_PIDは、オーディオストリームを伝送するトランスポートパケットのP I Dを示す。PTS\_EP\_startは、オーディオストリームのアクセスユニットのP T Sを示す。RSPN\_EP\_startは、AVストリームの中でPTS\_EP\_startで参照されるアクセスユニットの第1バイト目を含むソースポケットのアドレスを示す。

EP\_map\_for\_one\_stream\_PID()と呼ばれるサブテーブルは、同じP I Dを持つトランスポートパケットによって伝送されるオーディオストリーム毎に作られる。Clipの中に複数のオーディオストリームが存在する場合、EP\_mapは複数のEP\_map\_for\_one\_stream\_PID()を含んでもよい。

EP\_mapとSTC\_Infoの関係を説明すると、1つのEP\_map\_for\_one\_stream\_PID()は、STCの不連続点に関係なく1つのテーブルに作られる。RSPN\_EP\_startの値とSTC\_Info()において定義されるRSPN\_STC\_startの値を比較することにより、それぞれのSTC\_sequenceに属するEP\_mapのデータの境界が分かる（図68を参照）。EP\_mapは、同じPIDで伝送される連続したストリームの範囲に対して、1つのEP\_map\_for\_one\_stream\_PIDを持たねばならない。図69に示したような場合、program#1とprogram#3は、同じビデオPIDを持つが、データ範囲が連続していないので、それぞれのプログラム毎にEP\_map\_for\_one\_stream\_PIDを持たねばならない。

図70は、EP\_mapのシンタクスを示す図である。図70に示したEP\_mapのシンタクスを説明すると、EP\_typeは、4ビットのフィールドであり、図71に示すように、EP\_mapのエントリポイントタイプを示す。EP\_typeは、このフィールドに続くデータフィールドのセマンティクスを示す。Clipが1つ以上のビデオストリームを含む場合、EP\_typeは0('video')にセットされなければならない。又は、Clipがビデオストリームを含まず、1つ以上のオーディオストリームを含む場合、EP\_typeは1('audio')にセットされなければならない。

number\_of\_stream\_PIDsの16ビットのフィールドは、EP\_map()の中のnumber\_of\_stream\_PIDsを変数にもつfor-loopのループ回数を示す。stream\_PID(k)の16ビットのフィールドは、EP\_map\_for\_one\_stream\_PID(num\_EP\_entries(k))によって参照されるk番目のエレメンタリーストリーム（ビデオ又はオーディオストリーム）を伝送するトランスポートパケットのPIDを示す。EP\_typeが0('video')に等しい場合、そのエレメンタリーストリームはビデオストリームでなければならない。また、EP\_typeが1('audio')に等しい場合、そのエレメンタリーストリームはオーディオストリームでなければならない。

num\_EP\_entries(k)の16ビットのフィールドは、EP\_map\_for\_one\_stream\_PID(num\_EP\_entries(k))によって参照されるnum\_EP\_entries(k)を示す。EP\_map\_for\_one\_stream\_PID\_Start\_address(k): この32ビットのフィールドは、EP\_map()の中でEP\_map\_for\_one\_stream\_PID(num\_EP\_entries(k))が始まる相対バイト位置を示す。この値は、EP\_map()の第1バイト目からの大きさで示される。

`padding_word`は、`EP_map()`のシンタクスに従って挿入されなければならない。`X`と`Y`は、0又は任意の正の整数でなければならない。それぞれのパディングワードは、任意の値をとってもよい。

図72は、`EP_map_for_one_stream_PID`のシンタクスを示す図である。図72に示した`EP_map_for_one_stream_PID`のシンタクスを説明すると、`PTS_EP_start`の32ビットのフィールドのセマンティクスは、`EP_map()`において定義される`EP_type`により異なる。`EP_type`が0 ('video')に等しい場合、このフィールドは、ビデオストリームのシーケンスヘッダで始まるアクセスユニットの33ビット精度のPTSの上位32ビットを持つ。`EP_type`が1 ('audio')に等しい場合、このフィールドは、オーディオストリームのアクセスユニットの33ビット精度のPTSの上位32ビットを持つ。

`RSPN_EP_start`の32ビットのフィールドのセマンティクスは、`EP_map()`において定義される`EP_type`により異なる。`EP_type`が0 ('video')に等しい場合、このフィールドは、AVストリームの中で`PTS_EP_start`により参照されるアクセスユニットのシーケンスヘッダの第1バイト目を含むソースポケットの相対アドレスを示す。又は、`EP_type`が1 ('audio')に等しい場合、このフィールドは、AVストリームの中で`PTS_EP_start`により参照されるアクセスユニットのオーディオフレームの第一バイト目を含むソースポケットの相対アドレスを示す。

`RSPN_EP_start`は、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットから`ClipInfo()`において定義される`offset_SPN`の値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、

`SPN_xxx = RSPN_xxx - offset_SPN`

により算出される。シンタクスのfor-loopの中で`RSPN_EP_start`の値は、昇順に現れなければならない。

次に、`TU_map`について、図73を参照して説明する。`TU_map`は、ソースパケットのアライバルタイムクロック（到着時刻ベースの時計）に基づいて、1つの時間軸を作る。その時間軸は、`TU_map_time_axis`と呼ばれる。`TU_map_time_axis`の原点は、`TU_map()`の中の`offset_time`によって示される。`TU_map_time_axis`は、0

ffset\_timeから一定の単位に分割される。その単位を、time\_unitという。

AVストリームの中の各々のtime\_unitの中で、最初の完全な形のソースパケットのAVストリームファイル上のアドレスが、TU\_mapにストアされる。これらのアドレスを、RSPN\_time\_unit\_startという。TU\_map\_time\_axis上において、k (k >=0)番目のtime\_unitが始まる時刻は、TU\_start\_time(k)と呼ばれる。この値は次式に基づいて算出される。

$$TU_{start\_time}(k) = offset\_time + k * time\_unit\_size$$

TU\_start\_time(k)は、45MHzの精度を持つ。

図75は、TU\_mapのシンタクスを示す図である。図75に示したTU\_mapのシンタクスを説明すると、offset\_timeの32bit長のフィールドは、TU\_map\_time\_axisに対するオフセットタイムを与える。この値は、Clipの中の最初のtime\_unitに対するオフセット時刻を示す。offset\_timeは、27MHz精度のアライバルタイムクロックから導き出される45kHzクロックを単位とする大きさである。AVストリームが新しいClipとして記録される場合、offset\_timeは0にセットされなければならない。

time\_unit\_sizeの32ビットフィールドは、time\_unitの大きさを与えるものであり、それは27MHz精度のアライバルタイムクロックから導き出される45MHzクロックを単位とする大きさである。time\_unit\_sizeは、1秒以下 (time\_unit\_size <= 45000) にすることがよい。number\_of\_time\_unit\_entriesの32ビットフィールドは、TU\_map()の中にストアされているtime\_unitのエントリ数を示す。

RSPN\_time\_unit\_startの32ビットフィールドは、AVストリームの中でそれのtime\_unitが開始する場所の相対アドレスを示す。RSPN\_time\_unit\_startは、ソースパケット番号を単位とする大きさであり、AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset\_SPNの値を初期値としてカウントされる。そのAV streamファイルの中での絶対アドレスは、

$$SPN_{xxx} = RSPN_{xxx} - offset\_SPN$$

により算出される。シンタクスのfor-loopの中でRSPN\_time\_unit\_startの値は、昇順に現れなければならない。(k+1)番目のtime\_unitの中にソースパケット

が何もない場合、(k + 1)番目のRSPN\_time\_unit\_startは、k番目のRSPN\_time\_unit\_startと等しくなければならない。

図45に示したzzzzz.clipのシンタクス内のClipMarkについて説明する。ClipMarkは、クリップについてのマーク情報であり、ClipMarkの中にストアされる。このマークは、記録器（記録再生装置1）によってセットされるものであり、ユーザによってセットされるものではない。

図75は、ClipMarkのシンタクスを示す図である。図75に示したClipMarkのシンタクスを説明すると、version\_numberは、このClipMark()のバージョンナンバを示す4個のキャラクター文字である。version\_numberは、ISO 646に従って、"0045"と符号化されなければならない。

lengthは、このlengthフィールドの直後からClipMark()の最後までのClipMark()のバイト数を示す32ビットの符号なし整数である。number\_of\_Clip\_marksは、ClipMarkの中にストアされているマークの個数を示す16ビットの符号なし整数であり、number\_of\_Clip\_marksは、0であってもよい。mark\_typeは、マークのタイプを示す8ビットのフィールドであり、図76に示すテーブルに従って符号化される。

mark\_time\_stampは、32ビットフィールドであり、マークが指定されたポイントを示すタイムスタンプをストアする。mark\_time\_stampのセマンティクスは、図77に示すように、PlayList()の中のCPI\_typeにより異なる。

STC\_sequence\_idは、CPI()の中のCPI\_typeがEP\_map\_typeを示す場合、この8ビットのフィールドは、マークが置かれているところのSTC連続区間のSTC\_sequence\_idを示す。CPI()の中のCPI\_typeがTU\_map\_typeを示す場合、この8ビットのフィールドは何も意味を持たず、0にセットされる。character\_setの8ビットのフィールドは、mark\_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示される値に対応する。

name\_lengthの8ビットフィールドは、Mark\_nameフィールドの中に示されるマーク名のバイト長を示す。mark\_nameのフィールドは、マークの名称を示す。このフィールドの中の左からname\_length数のバイト数が、有効なキャラクター文字であり、それはマークの名称を示す。mark\_nameフィールドの中で、それら有効なキ

キャラクター文字の後の値は、どんな値が入っていてもよい。

`ref_thumbnail_index`のフィールドは、マークに付加されるサムネイル画像の情報を示す。`ref_thumbnail_index`フィールドが、`0xFFFF`でない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、`mark.thmb`ファイルの中にストアされている。その画像は、`mark.thmb`ファイルの中で`ref_thumbnail_index`の値を用いて参照される。`ref_thumbnail_index`フィールドが、`0xFFFF`である場合、そのマークにはサムネイル画像が付加されていない。

`MakersPrivateData`については、図22を参照して既に説明したので、その説明は省略する。

次に、サムネイルインフォメーション (Thumbnail Information) について説明する。サムネイル画像は、`menu.thmb`ファイル又は`mark.thmb`ファイルにストアされる。これらのファイルは同じシンタクス構造であり、ただ1つの`Thumbnail()`を持つ。`menu.thmb`ファイルは、メニューサムネイル画像、すなわち`Volume`を代表する画像、及び、それぞれの`PlayList`を代表する画像をストアする。全てのメニューサムネイルは、ただ1つの`menu.thmb`ファイルにストアされる。

`mark.thmb`ファイルは、マークサムネイル画像、すなわちマーク点を表すピクチャをストアする。全ての`PlayList`及び`Clip`に対する全てのマークサムネイルは、ただ1つの`mark.thmb`ファイルにストアされる。サムネイルは頻繁に追加、削除されるので、追加操作と部分削除の操作は容易に高速に実行できなければならない。この理由のため、`Thumbnail()`はブロック構造を有する。画像のデータはいくつかの部分に分割され、各部分は1つの`tn_block`に格納される。1つの画像データは連続した`tn_block`に格納される。`tn_block`の列には、使用されていない`tn_block`が存在してもよい。1つのサムネイル画像のバイト長は可変である。

図78は、`menu.thmb`と`mark.thmb`のシンタクスを示す図であり、図79は、図78に示した`menu.thmb`と`mark.thmb`のシンタクス内の`Thumbnail`のシンタクスを示す図である。図79に示した`Thumbnail`のシンタクスについて説明すると、`version_number`は、この`Thumbnail()`のバージョンナンバを示す4個のキャラクター文字である。`version_number`は、ISO 646に従って、"0045"と符号化されなければならない。

lengthは、このlengthフィールドの直後からThumbnail()の最後までのMakersPrivateData()のバイト数を示す32ビットの符号なし整数である。tn\_blocks\_start\_addressは、Thumbnail()の先頭のバイトからの相対バイト数を単位として、最初のtn\_blockの先頭バイトアドレスを示す32ビットの符号なし整数である。相対バイト数は0からカウントされる。number\_of\_thumbnailsは、Thumbnail()の中に含まれているサムネイル画像のエントリ数を与える16ビットの符号なし整数である。

tn\_block\_sizeは、1024バイトを単位として、1つのtn\_blockの大きさを与える16ビットの符号なし整数である。例えば、tn\_block\_size=1ならば、それは1つのtn\_blockの大きさが1024バイトであることを示す。number\_of\_tn\_blocksは、このThumbnail()中のtn\_blockのエントリ数を表す116ビットの符号なし整数である。thumbnail\_indexは、このthumbnail\_indexフィールドから始まるforループ一回分のサムネイル情報で表されるサムネイル画像のインデックス番号を表す16ビットの符号なし整数である。thumbnail\_indexとして、0xFFFFという値を使用してはならない。thumbnail\_indexはUIAppInfoVolume()、UIAppInfoPlayList()、PlayListMark()、及びClipMark()の中のref\_thumbnail\_indexによって参照される。

thumbnail\_picture\_formatは、サムネイル画像のピクチャフォーマットを表す8ビットの符号なし整数で、図80に示すような値をとる。表中のDCFとPNGは"menu.thmb"内でのみ許される。マークサムネイルは、値"0x00" (MPEG-2 Video I-picture)をとらなければならない。

picture\_data\_sizeは、サムネイル画像のバイト長をバイト単位で示す32ビットの符号なし整数である。start\_tn\_block\_numberは、サムネイル画像のデータが始まるtn\_blockのtn\_block番号を表す16ビットの符号なし整数である。サムネイル画像データの先頭は、tb\_blockの先頭と一致していかなければならない。tn\_block番号は、0から始まり、tn\_blockのfor-ループ中の変数kの値に関係する。

x\_picture\_lengthは、サムネイル画像のフレーム画枠の水平方向のピクセル数を表す16ビットの符号なし整数である。y\_picture\_lengthは、サムネイル画像のフレーム画枠の垂直方向のピクセル数を表す16ビットの符号なし整数である。

tn\_blockは、サムネイル画像がストアされる領域である。Thumbnail()の中の全てのtn\_blockは、同じサイズ（固定長）であり、その大きさはtn\_block\_sizeによって定義される。

図81A及び図81Bは、サムネイル画像データがどのようにtn\_blockに格納されるかを模式的に表した図である。図81A及び図81Bのように、各サムネイル画像データはtn\_blockの先頭から始まり、1 tn\_blockを超える大きさの場合は、連続する次のtn\_blockを使用してストアされる。このようにすることにより、可変長であるピクチャデータが、固定長のデータとして管理することが可能となり、削除といった編集に対して簡便な処理により対応することができるようになる。

次に、AVストリームファイルについて説明する。AVストリームファイルは、"M2TS"ディレクトリ（図14）にストアされる。AVストリームファイルには、2つのタイプがあり、それらは、Clip AVストリームとBridge-Clip AVストリームファイルである。両方のAVストリーム共に、これ以降で定義されるDVR MPEG-2トランSPORTストリームファイルの構造でなければならない。

先ず、DVR MPEG-2トランSPORTストリームについて説明する。DVR MPEG-2トランSPORTストリームの構造は、図82に示すようになっている。AVストリームファイルは、DVR MPEG 2トランSPORTストリームの構造を持つ。DVR MPEG 2トランSPORTストリームは、整数個のAligned unitから構成される。Aligned unitの大きさは、6144 バイト (2048 \* 3 バイト)である。Aligned unitは、ソースパケットの第1バイト目から始まる。ソースパケットは、192バイト長である。1つのソースパケットは、TP\_extra\_headerとトランSPORTパケットからなる。TP\_extra\_headerは、4バイト長であり、またトランSPORTパケットは、188バイト長である。

1つのAligned unitは、32個のソースパケットからなる。DVR MPEG 2トランSPORTストリームの中の最後のAligned unitも、また32個のソースパケットからなる。よって、DVR MPEG 2トランSPORTストリームは、Aligned unitの境界で終端する。ディスクに記録される入力トランSPORTストリームのトランSPORTパケットの数が32の倍数でないとき、ヌルパケット (PID=

0x1FFFのトランSPORTパケット)を持ったソースパケットを最後のAligned unitに使用しなければならない。ファイルシステムは、DVR MPEG 2トランSPORTストリームに余分な情報を付加してはならない。

図83に、DVR MPEG-2トランSPORTストリームのレコーダモデルを示す。図83に示したレコーダは、レコーディングプロセスを規定するための概念上のモデルである。DVR MPEG-2トランSPORTストリームは、このモデルに従う。

MPEG-2トランSPORTストリームの入力タイミングについて説明する。入力MPEG 2トランSPORTストリームは、フルトランSPORTストリーム又はパーシャルトランSPORTストリームである。入力されるMPEG 2トランSPORTストリームは、ISO/IEC 13818-1又はISO/IEC 13818-9に従っていなければならない。MPEG 2トランSPORTストリームのi番目のバイトは、T-STD (ISO/IEC 13818-1で規定されるTransport stream system target decoder)とソースパケッタイザへ、時刻t(i)に同時に入力される。Rpkは、トランSPORTパケットの入力レートの瞬時的な最大値である。

27MHz PLL52は、27MHzクロックの周波数を発生する。27MHzクロックの周波数は、MPEG-2トランSPORTストリームのPCR (Program Clock Reference)の値にロックされる。arrival time clock counter53は、27MHzの周波数のパルスをカウントするバイナリーカウンターである。Arrival\_time\_clock(i)は、時刻t(i)におけるArrival time clock counterのカウント値である。

source packetizer54は、全てのトランSPORTパケットにTP\_extra\_headerを付加し、ソースパケットを作る。Arrival\_time\_stampは、トランSPORTパケットの第1バイト目がT-STDとソースパケッタイザの両方へ到着する時刻を表す。Arrival\_time\_stamp(k)は、次式で示されるようにArrival\_time\_clock(k)のサンプル値であり、ここで、kはトランSPORTパケットの第1バイト目を示す。

$$\text{arrival\_time\_stamp}(k) = \text{arrival\_time\_clock}(k) \% 230$$

2つの連続して入力されるトランSPORTパケットの時間間隔が、230/2700

0000秒（約40秒）以上になる場合、その2つのトランSPORTパケットのarrive1\_time\_stampの差分は、 $230/27000000$ 秒になるようにセットされるべきである。レコーダは、そのようになる場合に備えてある。

smoothing buffer 5 5は、入力トランSPORTストリームのビットレートをスマージングする。スマージングバッファは、オーバーフローしてはならない。Rmaxは、スマージングバッファが空でないときのスマージングバッファからのソースパケットの出力ビットレートである。スマージングバッファが空であるとき、スマージングバッファからの出力ビットレートは0である。

次に、DVR MPEG-2トランSPORTストリームのレコーダモデルのパラメータについて説明する。Rmaxという値は、AVストリームファイルに対応するClipInfo()において定義されるTS\_recording\_rateによって与えられる。この値は、次式により算出される。

$$R_{max} = TS\_recording\_rate * 192/188$$

TS\_recording\_rateの値は、bytes/secondを単位とする大きさである。

入力トランSPORTストリームがSESFトランSPORTストリームの場合、Rpkは、AVストリームファイルに対応するClipInfo()において定義されるTS\_recording\_rateに等しくなければならない。入力トランSPORTストリームがSESFトランSPORTストリームでない場合、この値はMPEG-2 transport streamのデスクリプター、例えばmaximum\_bitrate\_descriptorやpartial\_transport\_stream\_descriptor等、において定義される値を参照してもよい。

smoothing buffer sizeは、入力トランSPORTストリームがSESFトランSPORTストリームの場合、スマージングバッファの大きさは0である。入力トランSPORTストリームがSESFトランSPORTストリームでない場合、スマージングバッファの大きさはMPEG-2 transport streamのデスクリプター、例えばsmoothing\_buffer\_descriptor、short\_smoothing\_buffer\_descriptor、partial\_transport\_stream\_descriptor等において定義される値を参照してもよい。

記録機（レコーダ）及び再生機（プレーヤ）は、十分なサイズのバッファを用意しなければならない。デフォルトのバッファサイズは、1536 bytesである。

次に、DVR MPEG-2トランSPORTストリームのプレーヤモデルについて

て説明する。図84は、DVR MPEG-2トランSPORTストリームのプレーヤモデルを示す図である。これは、再生プロセスを規定するための概念上のモデルである。DVR MPEG-2トランSPORTストリームは、このモデルに従う。

27MHz X-tal 61は、27MHzの周波数を発生する。27MHz周波数の誤差範囲は、 $+-30\text{ ppm}$  ( $27000000 \pm 810\text{ Hz}$ )でなければならない。arrival time clock counter 62は、27MHzの周波数のパルスをカウントするバイナリーカウンターである。Arrival\_time\_clock(i)は、時刻  $t(i)$  におけるArrival time clock counterのカウント値である。

smoothing buffer 64において、Rmaxは、スムージングバッファがフルでないときのスムージングバッファへのソースパケットの入力ビットレートである。スムージングバッファがフルであるとき、スムージングバッファへの入力ビットレートは0である。

MPEG-2トランSPORTストリームの出力タイミングを説明すると、現在のソースパケットのarrival\_time\_stampがarrival\_time\_clock(i)のLSB 30ビットの値と等しいとき、そのソースパケットのトランSPORTパケットは、スムージングバッファから引き抜かれる。Rpkは、トランSPORTパケットレートの瞬時的な

最大値である。スムージングバッファは、アンダーフロウしてはならない。

DVR MPEG-2トランSPORTストリームのプレーヤモデルのパラメータについては、上述したDVR MPEG-2トランSPORTストリームのレコーダモデルのパラメータと同一である。

図85は、Source packetのシンタクスを示す図である。transport\_packet()は、ISO/IEC 13818-1で規定されるMPEG-2トランSPORTパケットである。図85に示したSource packetのシンタクス内のTP\_Extra\_headerのシンタクスを図86に示す。図86に示したTP\_Extra\_headerのシンタクスについて説明すると、copy\_permission\_indicatorは、トランSPORTパケットのペイロードのコピー制限を表す整数である。コピー制限は、copy free、no more copy、copy once、又はcopy prohibitedとすることができます。図87は、copy\_permission\_indicatorの値と、それらによって指定されるモードの関係を示す。

`copy_permission_indicator`は、全てのトランスポートパケットに付加される。IEEE1394ディジタルインターフェースを使用して入力トランスポートストリームを記録する場合、`copy_permission_indicator`の値は、IEEE1394 isochronous packet headerの中のEMI (Encryption Mode Indicator)の値に関連付けてもよい。IEEE1394ディジタルインターフェースを使用しないで入力トランスポートストリームを記録する場合、`copy_permission_indicator`の値は、トランスポートパケットの中に埋め込まれたCCIの値に関連付けてもよい。アナログ信号入力をセルフエンコードする場合、`copy_permission_indicator`の値は、アナログ信号のCGMS-Aの値に関連付けてもよい。

`arrival_time_stamp`は、次式

$$\text{arrival\_time\_stamp}(k) = \text{arrival\_time\_clock}(k) \% 230$$

において、`arrival_time_stamp`によって指定される値を持つ整数値である。

Clip AVストリームの定義をするに、Clip AVストリームは、上述したような定義がされるDVR MPEG-2トランスポートストリームの構造を持たねばならない。`arrival_time_clock(i)`は、Clip AVストリームの中で連続して増加しなければならない。Clip AVストリームの中にシステムタイムベース (STCベース) の不連続点が存在したとしても、そのClip AVストリームの`arrival_time_clock(i)`は、連続して増加しなければならない。

Clip AVストリームの中の開始と終了の間の`arrival_time_clock(i)`の差分の最大値は、26時間でなければならない。この制限は、MPEG2トランスポートストリームの中にシステムタイムベース (STCベース) の不連続点が存在しない場合に、Clip AVストリームの中で同じ値のPTS (Presentation Time Stamp) が決して現れないことを保証する。MPEG2システムズ規格は、PTSのラップアラウンド周期を233/90000秒(約26.5時間)と規定している。

Bridge-Clip AVストリームの定義をするに、Bridge-Clip AVストリームは、上述したような定義がされるDVR MPEG-2トランスポートストリームの構造を持たねばならない。Bridge-Clip AVストリームは、1つのアライバルタイムベースの不連続点を含まなければならない。アライバルタイムベースの不連続点の前後のトランスポートストリームは、後述する符号化の制限に従わなければ

ならず、且つ後述するDVR-STDに従わなければならない。

本実施例においては、編集におけるPlayItem間のビデオとオーディオのシームレス接続をサポートする。PlayItem間をシームレス接続にすることは、プレーヤ／レコーダに”データの連続供給”と”シームレスな復号処理”を保証する。”

データの連続供給”とは、ファイルシステムが、デコーダにバッファのアンダーフロウを起こさせることのないように必要なビットレートでデータを供給することを保証できることである。データのリアルタイム性を保証して、データをディスクから読み出すことができるよう、データが十分な大きさの連続したブロック単位でストアされるようにする。

”シームレスな復号処理”とは、プレーヤが、デコーダの再生出力にボーズやギャップを起こさせることなく、ディスクに記録されたオーディオビデオデータを表示できることである。

シームレス接続されているPlayItemが参照するAVストリームについて説明する。先行するPlayItemと現在のPlayItemの接続が、シームレス表示できるように保証されているかどうかは、現在のPlayItemにおいて定義されているconnection\_conditionフィールドから判断することができる。PlayItem間のシームレス接続は、Bridge-Clipを使用する方法と使用しない方法がある。

図88は、Bridge-Clipを使用する場合の先行するPlayItemと現在のPlayItemの関係を示している。図88においては、プレーヤが読み出すストリームデータが、影を付けて示されている。図88に示したTS1は、Clip1 (Clip AVストリーム) の影を付けられたストリームデータとBridge-ClipのRSPN\_arrival\_time\_discontinuityより前の影を付けられたストリームデータからなる。

TS1のClip1の影を付けられたストリームデータは、先行するPlayItemのIN\_time (図88においてIN\_time1で図示されている) に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスから、RSPN\_exit\_from\_previous\_Clipで参照されるソースパケットまでのストリームデータである。TS1に含まれるBridge-ClipのRSPN\_arrival\_time\_discontinuityより前の影を付けられたストリームデータは、Bridge-Clipの最初のソースパケットから、RSPN\_arrival\_time\_discontinuityで参照されるソースパケットの直前のソースパケッ

トまでのストリームデータである。

また、図88におけるTS2は、Clip2 (Clip AVストリーム) の影を付けられたストリームデータとBridge-ClipのRSPN\_arrival\_time\_discontinuity以後の影を付けられたストリームデータからなる。TS2に含まれるBridge-ClipのRSPN\_arrival\_time\_discontinuity以後の影を付けられたストリームデータは、RSPN\_arrival\_time\_discontinuityで参照されるソースパケットから、Bridge-Clipの最後のソースパケットまでのストリームデータである。TS2のClip2の影を付けられたストリームデータは、RSPN\_enter\_to\_current\_Clipで参照されるソースパケットから、現在のPlayItemのOUT\_time (図88においてOUT\_time2で図示されている) に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスまでのストリームデータである。

図89は、Bridge-Clipを使用しない場合の先行するPlayItemと現在のPlayItemの関係を示している。この場合、プレーヤが読み出すストリームデータは、影を付けて示されている。図89におけるTS1は、Clip1 (Clip AVストリーム) の影を付けられたストリームデータからなる。TS1のClip1の影を付けられたストリームデータは、先行するPlayItemのIN\_time (図89においてIN\_time1で図示されている) に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスから始まり、Clip1の最後のソースパケットまでのデータである。

また、図89におけるTS2は、Clip2 (Clip AVストリーム) の影を付けられたストリームデータからなる。

TS2のClip2の影を付けられたストリームデータは、Clip2の最初のソースパケットから始まり、現在のPlayItemのOUT\_time (図89においてOUT\_time2で図示されている) に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスまでのストリームデータである。

図88と図89において、TS1とT2は、ソースパケットの連続したストリームである。次に、TS1とTS2のストリーム規定と、それらの間の接続条件について考える。先ず、シームレス接続のための符号化制限について考える。トランスポートストリームの符号化構造の制限として、先ず、TS1とTS2の中に

含まれるプログラムの数は、1でなければならない。TS1とTS2の中に含まれるビデオストリームの数は、1でなければならない。TS1とTS2の中に含まれるオーディオストリームの数は、2以下でなければならない。TS1とTS2の中に含まれるオーディオストリームの数は、等しくなければならない。TS1及び/又はTS2の中に、上記以外のエレメンタリーストリーム又はプライベートストリームが含まれていてもよい。

ビデオビットストリームの制限について説明する。図90は、ピクチャの表示順序で示すシームレス接続の例を示す図である。接続点においてビデオストリームをシームレスに表示できるためには、OUT\_time1 (Clip1のOUT\_time) の後とIN\_time2 (Clip2のIN\_time) の前に表示される不必要なピクチャは、接続点付近のClipの部分的なストリームを再エンコードするプロセスにより、除去されなければならない。

図90に示したような場合において、BridgeSequenceを使用してシームレス接続を実現する例を、図91に示す。RSPN\_arrival\_time\_discontinuityより前のBridge-Clipのビデオストリームは、図90のClip1のOUT\_time1に対応するピクチャまでの符号化ビデオストリームからなる。そして、そのビデオストリームは先行するClip1のビデオストリームに接続され、1つの連続でMPEG2規格に従ったエレメンタリーストリームとなるように再エンコードされている。

同様にして、RSPN\_arrival\_time\_discontinuity以後のBridge-Clipのビデオストリームは、図90のClip2のIN\_time2に対応するピクチャ以後の符号化ビデオストリームからなる。そして、そのビデオストリームは、正しくデコード開始することができて、これに続くClip2のビデオストリームに接続され、1つの連続でMPEG2規格に従ったエレメンタリーストリームとなるように再エンコードされている。Bridge-Clipを作るためには、一般に、数枚のピクチャは再エンコードしなければならず、それ以外のピクチャはオリジナルのClipからコピーすることができる。

図90に示した例の場合にBridgeSequenceを使用しないでシームレス接続を実現する例を図92に示す。Clip1のビデオストリームは、図90のOUT\_time1に対応するピクチャまでの符号化ビデオストリームからなり、それは、1つの連続で

MPEG 2 規格に従ったエレメンタリーストリームとなるように再エンコードされている。同様にして、Clip2のビデオストリームは、図 90 の Clip2 の IN\_time2 に対応するピクチャ以後の符号化ビデオストリームからなり、それは、1つの連続で MPEG 2 規格に従ったエレメンタリーストリームとなるように再エンコードされている。

ビデオストリームの符号化制限について説明すると、先ず、TS1 と TS2 のビデオストリームのフレームレートは、等しくなければならない。TS1 のビデオストリームは、sequence\_end\_code で終端しなければならない。TS2 のビデオストリームは、Sequence Header、GOP Header、そして I-ピクチャで開始しなければならない。TS2 のビデオストリームは、クローズド GOP で開始しなければならない。

ビットストリームの中で定義されるビデオプレゼンテーションユニット（フレーム又はフィールド）は、接続点を挟んで連続でなければならない。接続点において、フレーム又はフィールドのギャップがあつてはならない。接続点において、トップボトムのフィールドシーケンスは連続でなければならない。3-2 ブルダウントップボトムのフィールドシーケンスは連続でなければならない。"top\_field\_first" 及び "repeat\_first\_field" フラグを書き換える必要があるかもしれない、又はフィールドギャップの発生を防ぐために局所的に再エンコードするようにしてもよい。

オーディオビットストリームの符号化制限について説明すると、TS1 と TS2 のオーディオのサンプリング周波数は、同じでなければならない。TS1 と TS2 のオーディオの符号化方法（例. MPEG1レイヤ2, AC-3, SESA PCM, AAC）は、同じでなければならない。

次に、MPEG-2 トランスポートストリームの符号化制限について説明すると、TS1 のオーディオストリームの最後のオーディオフレームは、TS1 の最後の表示ピクチャの表示終了時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。TS2 のオーディオストリームの最初のオーディオフレームは、TS2 の最初の表示ピクチャの表示開始時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。

接続点において、オーディオプレゼンテーションユニットのシーケンスにギャ

ップがあつてはならない。図93に示すように、2オーディオフレーム区間未満のオーディオプレゼンテーションユニットの長さで定義されるオーバーラップがあつてもよい。TS2のエレメンタリーストリームを伝送する最初のパケットは、ビデオパケットでなければならない。接続点におけるトランスポストストリームは、後述するDVR-STDに従わなくてはならない。

Clip及びBridge-Clipの制限について説明すると、TS1とTS2は、それの中にアライバルタイムベースの不連続点を含んではならない。

以下の制限は、Bridge-Clipを使用する場合にのみ適用される。TS1の最後のソースパケットとTS2の最初のソースパケットの接続点においてのみ、Bridge-Clip AVストリームは、ただ1つのアライバルタイムベースの不連続点を持つ。ClipInfo()において定義されるRSPN\_arrival\_time\_discontinuityが、その不連続点のアドレスを示し、それはTS2の最初のソースパケットを参照するアドレスを示さなければならない。

BridgeSequenceInfo()において定義されるRSPN\_exit\_from\_previous\_Clipによって参照されるソースパケットは、Clip1の中のどのソースパケットでもよい。それは、Aligned unitの境界である必要はない。BridgeSequenceInfo()において定義されるRSPN\_enter\_to\_current\_Clipによって参照されるソースパケットは、Clip2の中のどのソースパケットでもよい。それは、Aligned unitの境界である必要はない。

PlayItemの制限について説明すると、先行するPlayItemのOUT\_time (図88、図89において示されるOUT\_time1) は、TS1の最後のビデオプレゼンテーションユニットの表示終了時刻を示さなければならない。現在のPlayItemのIN\_time (図88、図89において示されるIN\_time2) は、TS2の最初のビデオプレゼンテーションユニットの表示開始時刻を示さなければならない。

Bridge-Clipを使用する場合のデータアロケーションの制限について、図94を参照して説明すると、シームレス接続は、ファイルシステムによってデータの連續供給が保証されるように作られなければならない。これは、Clip1 (Clip AVストリームファイル) とClip2 (Clip AVストリームファイル) に接続されるBridge-Clip AVストリームを、データアロケーション規定を満たすように配置す

ることによって行われなければならない。

RSPN\_exit\_from\_previous\_Clip以前のClip1 (Clip AVストリームファイル) のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているよう に、RSPN\_exit\_from\_previous\_Clipが選択されなければならない。Bridge-Clip AVストリームのデータ長は、ハーフフラグメント以上の連続領域に配置される ように、選択されなければならない。RSPN\_enter\_to\_current\_Clip以後のClip2 (Clip AVストリームファイル) のストリーム部分が、ハーフフラグメント以上 の連続領域に配置されているように、RSPN\_enter\_to\_current\_Clipが選択されなければなら ない。

Bridge-Clipを使用しないでシームレス接続する場合のデータアロケーションの 制限について、図95を参照して説明すると、シームレス接続は、ファイルシステム によってデータの連続供給が保証されるように作られなければならない。これは、Clip1 (Clip AVストリームファイル) の最後の部分とClip2 (Clip AVストリームファイル) の最初の部分を、データアロケーション規定を満たすよう に配置することによって行われなければならない。

Clip1 (Clip AVストリームファイル) の最後のストリーム部分が、ハーフフラグメント 以上の連続領域に配置されていなければならぬ。Clip2 (Clip AVストリームファイル) の最初のストリーム部分が、ハーフフラグメント以上の連 続領域に配置されていなければならぬ。

次に、DVR-STDについて説明する。DVR-STDは、DVR MPEG 2トランスポートストリームの生成及び検証の際ににおけるデコード処理をモデル 化するための概念モデルである。また、DVR-STDは、上述したシームレス 接続された2つのPlayItemによって参照されるAVストリームの生成及び検証の 際におけるデコード処理をモデル化するための概念モデルである。

DVR-STDモデルを図96に示す。図96に示したモデルには、DVR MPEG-2トランスポートストリームプレーヤモデルが構成要素として含まれて いる。n, TBn, MBn, EBn, TBsys, Bsys, Rxn, Rbxn, Rxsy, Dn, Dsys, On及びP n(k)の表記方法は、ISO/IEC13818-1のT-STDに定義されてい るものと同じである。すなわち、次の通りである。nは、エレメンタリーストリー

ムのインデクス番号である。TBnは、エレメンタリーストリームnのトランスポートバッファである。

MBnは、エレメンタリーストリームnの多重バッファである。ビデオストリームについてのみ存在する。EBnは、エレメンタリーストリームnのエレメンタリーストリームバッファである。ビデオストリームについてのみ存在する。TBsysは、復号中のプログラムのシステム情報のための入力バッファである。Bsysは、復号中のプログラムのシステム情報のためのシステムターゲットデコーダ内のメインバッファである。Rxnは、データがTBnから取り除かれる伝送レートである。Rbxnは、PESパケットペイロードがMBnから取り除かれる伝送レートである。ビデオストリームについてのみ存在する。

Rxsysは、データがTBsysから取り除かれる伝送レートである。Dnは、エレメンタリーストリームnのデコーダである。Dsysは、復号中のプログラムのシステム情報に関するデコーダである。Onは、ビデオストリームnのre-ordering bufferである。Pn(k)は、エレメンタリーストリームnのk番目のプレゼンテーションユニットである。

DVR-STDのデコーディングプロセスについて説明する。単一のDVR-MPEG-2トランSPORTストリームを再生している間は、トランSPORTパケットをTB1, TBn又はTBsysのバッファへ入力するタイミングは、ソースパケットのarrival\_time\_stampにより決定される。TB1, MB1, EB1, TBn, Bn, TBsys及びBsysのバッファリング動作の規定は、ISO/IEC 13818-1に規定されているT-STDと同じである。復号動作と表示動作の規定もまた、ISO/IEC 13818-1に規定されているT-STDと同じである。

シームレス接続されたPlayItemを再生している間のデコーディングプロセスについて説明する。ここでは、シームレス接続されたPlayItemによって参照される2つのAVストリームの再生について説明をすることにし、以後の説明では、上述した（例えば、図88に示した）TS1とTS2の再生について説明する。TS1は、先行するストリームであり、TS2は、現在のストリームである。

図97は、あるAVストリーム（TS1）からそれにシームレスに接続された次のAVストリーム（TS2）へと移るときのトランSPORTパケットの入力、

復号、表示のタイミングチャートを示す。所定のAVストリーム(TS1)からそれにシームレスに接続された次のAVストリーム(TS2)へと移る間には、TS2のアライバルタイムベースの時間軸(図97においてATC2で示される)は、TS1のアライバルタイムベースの時間軸(図97においてATC1で示される)と同じでない。

また、TS2のシステムタイムベースの時間軸(図97においてSTC2で示される)は、TS1のシステムタイムベースの時間軸(図97においてSTC1で示される)と同じでない。ビデオの表示は、シームレスに連続していることが要求される。オーディオのプレゼンテーションユニットの表示時間にはオーバーラップがあつてもよい。

DVR-STDへの入力タイミングについて説明する。時刻T1までの時間、すなわち、TS1の最後のビデオパケットがDVR-STDのTB1に入力終了するまでは、DVR-STDのTB1、TBn又はTBsysのバッファへの入力タイミングは、TS1のソースパケットのarrival\_time\_stampによって決定される。

TS1の残りのパケットは、TS\_recording\_rate(TS1)のビットレートでDVR-STDのTBn又はTBsysのバッファへ入力されなければならない。ここで、TS\_recording\_rate(TS1)は、Clip1に対応するClipInfo()において定義されるTS\_recording\_rateの値である。TS1の最後のバイトがバッファへ入力する時刻は、時刻T2である。したがって、時刻T1からT2までの区間では、ソースパケットのarrival\_time\_stampは無視される。

N1をTS1の最後のビデオパケットに続くTS1のトランスポートパケットのバイト数とすると、時刻T1乃至T2までの時間DT1は、N1バイトがTS\_recording\_rate(TS1)のビットレートで入力終了するために必要な時間であり、次式により算出される。

$$DT1 = T2 - T1 = N1 / TS\_recording\_rate$$

(TS1)時刻T1乃至T2までの間は、RXnとRXsysの値は共に、TS\_recording\_rate(TS1)の値に変化する。このルール以外のバッファリング動作は、STDと同じである。

T2の時刻において、arrival time clock counterは、TS2の最初のソースパ

ケットのarrival\_time\_stampの値にリセットされる。DVR-STDのTB1, TBn 又はTBsysのバッファへの入力タイミングは、TS2のソースパケットのarrival\_time\_stampによって決定される。RXnとRXsysは共に、T-STDにおいて定義されている値に変化する。

附加的なオーディオバッファリング及びシステムデータバッファリングについて説明すると、オーディオデコーダとシステムデコーダは、時刻T1からT2までの区間の入力データを処理することができるよう、T-STDで定義されるバッファ量に加えて附加的なバッファ量（約1秒分のデータ量）が必要である。

ビデオのプレゼンテーションタイミングについて説明すると、ビデオプレゼンテーションユニットの表示は、接続点を通して、ギャップなしに連続でなければならない。ここで、STC1は、TS1のシステムタイムベースの時間軸（図97ではSTC1と図示されている）とし、STC2は、TS2のシステムタイムベースの時間軸（図97ではSTC2と図示されている。正確には、STC2は、TS2の最初のPCRがT-STDに入力した時刻から開始する。）とする。

STC1とSTC2の間のオフセットは、次のように決定される。PTS1endは、TS1の最後のビデオプレゼンテーションユニットに対応するSTC1上のPTSであり、PTS2startは、TS2の最初のビデオプレゼンテーションユニットに対応するSTC2上のPTSであり、Tppは、TS1の最後のビデオプレゼンテーションユニットの表示期間とすると、2つのシステムタイムベースの間のオフセットSTC\_deltaは、次式により算出される。

$$\text{STC\_delta} = \text{PTS1end} + \text{Tpp} - \text{PTS2start}$$

オーディオのプレゼンテーションのタイミングについて説明すると、接続点において、オーディオプレゼンテーションユニットの表示タイミングのオーバーラップがあつても良く、それは0乃至2オーディオフレーム未満である（図97に図示されている"audio overlap"を参照）。どちらのオーディオサンプルを選択するかということと、オーディオプレゼンテーションユニットの表示を接続点の後の補正されたタイムベースに再同期することは、プレーヤ側により設定されることである。

DVR-STDのシステムタイムクロックについて説明すると、時刻T5にお

いて、TS1の最後のオーディオプレゼンテーションユニットが表示される。システムタイムクロックは、時刻T2からT5の間にオーバーラップしていくてもよい。この区間では、DVR-STDは、システムタイムクロックを古いタイムベースの値(STC1)と新しいタイムベースの値(STC2)の間で切り替える。STC2の値は、次式により算出される。

$$STC2 = STC1 - STC_{\text{delta}}$$

バッファリングの連続性について説明する。STC11video\_endは、TS1の最後のビデオパケットの最後のバイトがDVR-STDのTB1へ到着するときのシステムタイムベースSTC1上のSTCの値である。STC22video\_startは、TS2の最初のビデオパケットの最初のバイトがDVR-STDのTB1へ到着するときのシステムタイムベースSTC2上のSTCの値である。STC21video\_endは、STC11video\_endの値をシステムタイムベースSTC2上の値に換算した値である。STC21video\_endは、次式により算出される。

$$STC21video_end = STC11video_end - STC_{\text{delta}}$$

DVR-STDに従うために、次の2つの条件を満たすことが要求される。まず、TS2の最初のビデオパケットのTB1への到着タイミングは、次に示す不等式を満たさなければならない。そして、次に示す不等式を満たさなければならない。

$$STC22video_start > STC21video_end + \Delta T1$$

この不等式が満たされるように、Clip1及び、又は、Clip2の部分的なストリームを再エンコード及び、又は、再多重化する必要がある場合は、その必要に応じて行われる。

次に、STC1とSTC2を同じ時間軸上に換算したシステムタイムベースの時間軸上において、TS1からのビデオパケットの入力とそれに続くTS2からのビデオパケットの入力は、ビデオバッファをオーバーフロウ及びアンダーフローさせてはならない。

このようなシンタクス、データ構造、規則に基づくことにより、記録媒体に記録されているデータの内容、再生情報等を適切に管理することができ、もって、ユーザが再生時に適切に記録媒体に記録されているデータの内容を確認したり、所望のデータを簡便に再生できるようにすることができる。

なお、以上の例は、多重化ストリームとしてM P E G 2 トランSPORTストリームを例にして説明しているが、これに限らず、M P E G 2 プログラムストリームや米国のD i r e c T V サービス（商標）で使用されているD S S トランSPORTストリームについても適用することが可能である。

次に、図98は、PlayListファイルの別の例を示す。図98と図23のシンタクスの大きな違いは、UIAppInfoPlayList()をストアしている場所である。図98の例では、UIAppInfoPlayList()がPlayList()の中から外に出されているので、UIAppInfoPlayList()の将来の情報拡張が比較的容易に行えるようになる。

version\_numberは、このサムネイルヘッダ情報ファイルのバージョンナンバを示す4個の数字である。

PlayList\_start\_addressは、PlayListファイルの先頭のバイトからの相対バイト数を単位として、PlayList()の先頭アドレスを示す。相対バイト数は0からカウントされる。

PlayListMark\_start\_addressは、PlayListファイルの先頭のバイトからの相対バイト数を単位として、PlayListMark()の先頭アドレスを示す。相対バイト数は0からカウントされる。

MakersPrivateData\_start\_addressは、PlayListファイルの先頭のバイトからの相対バイト数を単位として、MakersPrivateData()の先頭アドレスを示す。相対バイト数は0からカウントされる。

図99は、図98のPlayListファイルの中のUIAppInfoPlayListのシンタクスを示す。PlayList\_service\_typeは、PlayListファイルのタイプを示す。その一例は、図26に示されている。また、PlayList\_service\_typeは、ディジタルT V放送のプログラムが示すサービスタイプと同じ意味を持たせてもよい。例えば、日本のディジタルB S放送の場合、サービスタイプは、テレビサービス、音声サービス、及びデータ放送サービスの3種類を持つ。PlayListが使用するClip A Vストリームが含むプログラムのサービスタイプを代表する値をPlayList\_service\_typeにセットする。

PlayList\_character\_setは、channel\_name, PlayList\_name及びPlayList\_data ilフィールドに符号化されているキャラクター文字の符号化方法を示す。また、

これはPlayListMarkの中のmark\_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。

channel\_numberは、そのPlayListが記録されるとき、ユーザによって選択された放送チャンネル番号又はサービス番号を示す。複数のPlayListが1つのPlayListにコンパインされた場合は、このフィールドはそのPlayListの代表値を示す。このフィールドが0xFFFFにセットされている場合、このフィールドは何も意味を持たない。

channel\_name\_lengthは、channel\_nameフィールドの中に示されるチャンネル名のバイト長を示す。このフィールドは、20以下の値である。

channel\_nameは、そのPlayListが記録されるとき、ユーザによって選択された放送チャンネル又はサービスの名前を示す。このフィールドの中の左からchannel\_name\_lengthによって示される数のバイト数が有効なキャラクター文字であり、前記名前を示す。このフィールドの中で、それら有効なキャラクター文字に続く残りのバイトは、どんな値がセットされていてもよい。複数のPlayListが1つのPlayListにコンパインされた場合は、このフィールドはそのPlayListを代表する名前を示す。

PlayList\_name\_lengthは、PlayList\_nameフィールドの中に示されるPlayList名のバイト長を示す。

PlayList\_nameは、PlayListの名前を示す。このフィールドの中の左からPlayList\_name\_lengthによって示される数のバイト数が有効なキャラクター文字であり、前記名前を示す。このフィールドの中で、それら有効なキャラクター文字に続く残りのバイトは、どんな値がセットされていてもよい。

PlayList\_detail\_lengthは、PlayList\_detailフィールドの中に示されるPlayListの詳細情報のバイト長を示す。このフィールドは、1200以下の値である。

PlayList\_detailは、PlayListの詳細情報を説明するテキストを示す。このフィールドの中の左からPlayList\_detail\_lengthによって示される数のバイト数が有効なキャラクター文字であり、前記テキストを示す。このフィールドの中で、それら有効なキャラクター文字に続く残りのバイトは、どんな値がセットされていてもよい。

これ以外のシンタクスフィールドの意味は、図27に示す同名のフィールドと同じである。

図100は、図98のPlayListファイルの中のPlayList()のシンタクスを示す。図25の例と比べると、UIAppInfoPlayList()がなくなった点が違うだけで、これ以外は基本的に同じである。

図101は、SubPlayItemのシンタクスの別例を示す。図40の例と比べると、STC\_sequence\_idが追加された点が大きな違いである。

STC\_sequence\_idは、Clip\_Information\_file\_nameに対応するAVストリームファイル上の再生区間を特定するためのSubPath\_IN\_timeとSubPath\_OUT\_timeが参照するところのSTCのSTC\_sequence\_idを示す。SubPath\_IN\_timeとSubPath\_OUT\_timeは、STC\_sequence\_idによって指定される同じSTC連続区間上の時間を示す。

SubPlayItemにSTC\_sequence\_idを追加することにより、SubPlayItemが参照するAVストリームファイルがSTC不連続点を持つことが許されるようになる。

これ以外のシンタクスフィールドの意味は、図40に示す同名のフィールドと同じである。

図102は、Real PlayListの作成方法を説明するフローチャートを示す。図1の記録再生装置のブロック図を参照しながら説明する。

ステップS11で、制御部23はClip AVストリームを記録する。

ステップS12で、制御部23はClip AVストリームのEP\_mapを作成可能どうかを調べる。ステップS12で、Yesの場合はステップS13へ進み、EP\_mapを作成する。ステップS12で、Noの場合はステップS14へ進み、TU\_mapを作成する。

その後、ステップS15で、制御部23はPlayListのCPI\_typeをセットする。

ステップS16で、制御部23は上記Clipの全ての再生可能範囲をカバーするPlayItemからなるPlayList()を作成する。CPI\_typeがEP\_mapタイプの場合は、時間情報をPTSベースでセットする、このとき、Clipの中にSTC不連続点があり、PlayList()が2つ以上のPlayItemからなる場合は、PlayItem間のconnection\_conditionもまた決定する。CPI\_typeがTU\_mapタイプの場合は、時間情報をアライバルタイムベースでセットする。

ステップS17で、制御部23はUIAppInfoPlayList()を作成する。

ステップS18で、制御部23はPlayListMarkを作成する。

ステップS19で、制御部23はMakersPrivateDataを作成する。

ステップS20で、制御部23はReal Playlistファイルを記録する。

このようにして、新規にClip AVストリームを記録する毎に、1つのReal Playlistファイルが作られる。

図103は、Virtual Playlistの作成方法を説明するフローチャートである。

ステップS31で、ユーザインタフェースを通して、ディスクに記録されている1つのReal Playlistが指定される。そして、そのReal Playlistの再生範囲の中から、ユーザインタフェースを通して、IN点とOUT点で示される再生区間が指定される。CPI\_typeがEP\_mapタイプの場合は、再生区間をPTSベースでセットし、CPI\_typeがTU\_mapタイプの場合は、再生区間をアライバルタイムベースでセットする。

ステップS32で、制御部23はユーザによる再生範囲の指定操作が全て終了したか調べる。ユーザが上記指示した再生区間に続けて再生する区間を選ぶ場合はステップS31へ戻る。ステップS32でユーザによる再生範囲の指定操作が全て終了した場合は、ステップS33へ進む。

ステップS33で、連続して再生される2つの再生区間の間の接続状態(connection\_condition)を、ユーザがユーザインタフェースを通して決定するか、又は制御部23が決定する。

ステップS34で、CPI\_typeがEP\_mapタイプの場合、ユーザインタフェースを通して、ユーザがサブバス(アフレコ用オーディオ)情報を指定する。ユーザがサブバスを作成しない場合はこのステップはない。

ステップS35で、制御部23はユーザが指定した再生範囲情報、及びconnection\_conditionに基づいて、PlayList()を作成する。

ステップS36で、制御部23はUIAppInfoPlayList()を作成する。

ステップS37で、制御部23はPlayListMarkを作成する。

ステップS38で、制御部23はMakersPrivateDataを作成する。

ステップS39で、制御部23はVirtual Playlistファイルを記録する。

このようにして、ディスクに記録されているReal PlayListの再生範囲の中から、ユーザが見たい再生区間を選択してその再生区間をグループ化したもの毎に、1つのVirtual PlayListファイルが作られる。

図104はPlayListの再生方法を説明するフローチャートである。

ステップS51で、制御部23はInfo.dvr, Clip Information file, PlayList file及びサムネイルファイルの情報を取得し、ディスクに記録されているPlayListの一覧を示すG U I画面を作成し、ユーザインターフェースを通して、G U Iに表示する。

ステップS52で、制御部23はそれぞれのPlayListのUIAppInfoPlayList()に基づいて、PlayListを説明する情報をG U I画面に提示する。

ステップS53で、ユーザインターフェースを通して、G U I画面上からユーザが1つのPlayListの再生を指示する。

ステップS54で、制御部23は、CPI\_typeがEP\_mapタイプの場合、現在のPlayItemのSTC-sequenc-idとIN\_timeのP T Sから、IN\_timeより時間的に前で最も近いエントリポイントのあるソースパケット番号を取得する。又は制御部23は、CPI\_typeがTU\_mapタイプの場合、現在のPlayItemのIN\_timeから、IN\_timeより時間的に前で最も近いタイムユニットの開始するソースパケット番号を取得する。

ステップS55で、制御部23は上記ステップで得られたソースパケット番号からAVストリームのデータを読み出し、AVデコーダ27へ供給する。

ステップS56で、現在のPlayItemの時間的に前のPlayItemがあった場合は、制御部23は、前のPlayItemと現在のPlayItemとの表示の接続処理をconnection\_conditionに従って行う。

ステップS57で、制御部23は、CPI\_typeがEP\_mapタイプの場合、AVデコーダ27は、IN\_timeのP T Sのピクチャから表示を開始するように指示する。又は、制御部23は、CPI\_typeがEP\_mapタイプの場合、AVデコーダ27は、IN\_time以後のストリームのピクチャから表示を開始するように指示する。

ステップS58で、制御部23は、AVデコーダ27にAVストリームのデコードを続けるように指示する。

ステップS59で、制御部23は、CPI\_typeがEP\_mapタイプの場合、現在表示

の画像が、OUT\_timeのPTSの画像かを調べる。又は、制御部23は、CPI\_typeがTU\_mapタイプの場合、現在デコードしているストリームがOUT\_timeを過ぎたかを調べる。

ステップS59で、Noの場合は、ステップS60へ進む。ステップS60で現在の画像を表示して、ステップS58へ戻る。Yesの場合は、ステップS61へ進む。

ステップS61で、制御部23は、現在のPlayItemがPlayListの中で最後のPlayItemかを調べる。Noの場合はステップS54へ戻る。Yesの場合は、PlayListの再生を終了する。

図105は、PlayListのSubバスの再生方法を説明するフローチャートである。図105のPlayListのサブバスの再生方法は、PlayListのCPI\_typeがEP\_mapの場合のみに用いられる。このフローチャートの処理は、図104のPlayListの再生におけるステップS54以後の処理と共に、同時に行われる。また、AVデコーダ27は同時に2本のオーディオストリームのデコードが可能であることを前提とする。

ステップS71で、制御部23は、SubPlayItemの情報を取得する。

ステップS72で、制御部23は、SubPath\_IN\_timeよりも時間的に前で最も近いエントリポイントのあるソースパケット番号を取得する。

ステップS73で、制御部23は、上記エントリポイントのあるソースパケット番号からサブバスのAVストリームのデータを読み出し、AVデコーダ27へ供給する。

ステップS74で、制御部23は、Mainバスの再生が、sync\_PlayItem\_idとsync\_start PTS\_of\_PlayItemで示されるピクチャになったら、サブバスのオーディオを表示を開始するようにAVデコーダ27に指示する。

ステップS75でAVデコーダ27は、サブバスのAVストリームのデコードを続ける。

ステップS76で制御部23は、現在表示するサブバスのPTSが、SubPath\_OUT\_timeかを調べる。Noの場合は、ステップS77へ進む。ステップS77でサブバスの表示を続けて、ステップS75へ戻る。

ステップS 7 6で現在表示するサブバスのPTSが、SubPath\_OUT\_timeの場合はサブバスの表示を終了する。

図104及び図105のようにして、ユーザにより再生指示された1つのPlayListファイルのメインバス及びサブバスの再生が行なわれる。

図106は、PlayListMarkの作成方法を説明するフローチャートを示す。図1の記録再生装置のブロック図を参照しながら説明する。

ステップS 9 1で、制御部23はInfo.dvr, Clip Information file, PlayList file及びThumbnail fileの情報を取得し、ディスクに記録されているPlayListの一覧を示すG U I画面を作成し、ユーザインタフェースを通して、G U Iに表示する。

ステップS 9 2で、ユーザインタフェースを通して、ユーザが1つのPlayListの再生を制御部23に指示する。

ステップS 9 3で、制御部23は、上記指示されたPlayListの再生を開始させる（図104参照）。

ステップS 9 4で、ユーザインタフェースを通して、ユーザがお気に入りのシーンのところにマークのセットを制御部23に指示する。

ステップS 9 5で、制御部23は、CPI\_typeがEP\_mapの場合、マークのPTSとそれが属するPlayItemのPlayItem\_idを取得する。又は制御部23は、CPI\_typeがTU\_mapの場合、マーク点のアライバルタイムを取得する。

ステップS 9 6で、制御部23はマークの情報をPlayListMark()にストアする。

ステップS 9 7で、制御部23は、PlayListファイルを記録媒体100に記録する。

図107は、PlayListMarkを使用した頭出し再生方法を説明するフローチャートである。図1の記録再生装置のブロック図を参照しながら説明する。

ステップS 1 1 1で、制御部23はInfo.dvr, Clip Information file, PlayList file及びThumbnail fileの情報を取得し、ディスク（記録媒体100）に記録されているPlayListの一覧を示すG U I画面を作成し、ユーザインタフェースを通して、G U Iに表示する。

ステップS 1 1 2で、制御部23は、ユーザインタフェースを通して、ユーザ

が 1 つ の PlayList の再生を指示する。

ステップ S 1 1 3 で、制御部 2 3 は PlayListMark で参照されるピクチャから生成したサムネイルのリストを、ユーザインタフェースを通して、G U I に表示する。

ステップ S 1 1 4 で、ユーザインタフェースを通して、制御部 2 3 にユーザが再生開始点のマーク点を指定する。

ステップ S 1 1 5 で、制御部 2 3 は、CPI\_type は EP\_map タイプの場合は、マークの P T S とそれが属する PlayItem\_id を取得する。又は制御部 2 3 は、CPI\_type は TU\_map タイプの場合は、マークの A T S (Arrival Time Stamp) を取得する。

ステップ S 1 1 6 で、制御部 2 3 は、CPI\_type は EP\_map タイプの場合、PlayItem\_id が指す PlayItem が参照する A V ストリームの STC-sequence-id を取得する。

ステップ S 1 1 7 で、制御部 2 3 は、CPI\_type は EP\_map タイプの場合は、上記 STC-sequence-id とマークの P T S に基づいて、A V ストリームをデコーダへ入力する。具体的には、この STC-sequence-id とマーク点の P T S を用いて、図 1 0 4 のステップ S 5 4 、ステップ S 5 5 と同様の処理を行う。又は、制御部 2 3 は、CPI\_type は TU\_map タイプの場合は、マークの A T S に基づいて、A V ストリームをデコーダへ入力する。具体的には、この A T S を用いて図 1 0 4 のステップ S 5 4 、ステップ S 5 5 と同様の処理を行う。

ステップ S 1 1 8 で、制御部 2 3 は、CPI\_type が EP\_map タイプの場合は、マーク点の P T S のピクチャから表示を開始させる。又は制御部 2 3 は、CPI\_type が TU\_map タイプの場合は、マーク点の A T S 以後のピクチャから表示を開始させる。

このように、図 1 0 6 のようにして、ユーザが PlayList からお気に入りのシンの開始点等を選び、それをレコーダ（記録再生装置 1 の制御部 2 3 ）は PlayListMark に管理する。また図 1 0 7 のようにして、ユーザが PlayListMark にストアされているマーク点のリストから再生開始点を選択して、プレーヤはその開始点から再生を開始する。

このようなシンタクス、データ構造、規則に基づくことにより、記録媒体に記録されているデータの内容、再生情報等を適切に管理することができ、もって、ユーザが再生時に適切に記録媒体に記録されているデータの内容を確認したり、

所望のデータを簡便に再生できるようにすることができる。

Iピクチャの位置を分析できる場合、EP\_mapを用い、Iピクチャの位置を分析できない場合、TU\_mapを用いるようにすることで、共通のアプリケーションプログラム（ソフトウェア）で、異なるフォーマットのAVストリームを、同一の記録媒体に対して記録し、再生し、管理することが可能となる。

AVストリームを、その中身（Iピクチャの位置）を分析して記録媒体に記録する場合（コグニザント記録する場合）、TU\_mapを使用し、その中身（Iピクチャの位置）を分析せずに、そのまま記録媒体に記録する場合（ノンコグニザント記録する場合）、EP\_mapを使用する等して、共通のアプリケーションプログラムで、AVデータを、同一の記録媒体に記録し、再生し、管理することができる。

したがって、例えば、スクランブルされたAVデータを、デスクランブルして（分析して）記録媒体に記録する場合、TU\_mapを使用し、デスクランブルせずに（分析せずに）、そのまま記録媒体に記録する場合、EP\_mapを使用する等して、共通のアプリケーションプログラムで、AVデータを、同一の記録媒体に記録し、再生し、管理することができる。

更に、EP\_map typeとTU\_map typeを、CPI\_typeとして、PlayList()中に、記述できるようにしたので、Iピクチャの位置が分析できる場合、EP\_mapを用い、Iピクチャの位置が分析できない場合、TU\_mapを用いるようにすることができる。これにより、Iピクチャの位置を分析して記録するAVストリームデータと、分析しないで記録するAVストリームデータを、フラグを設定するだけで、共通のプログラムにより、統一して管理することができる。

また、PlayListファイルやClip Informationファイルを別々に分離して記録するので、編集等によって、あるPlayListやClipの内容を変更したとき、そのファイルに関係のない他のファイルを変更する必要がない。したがって、ファイルの内容の変更が容易に行え、またその変更及び記録にかかる時間を小さくできる。

更に、最初にInfo.dvrだけを読み出して、ディスクの記録内容をユーザインターフェースへ提示し、ユーザが再生指示したPlayListファイルとそれに関連するClip Informationファイルだけをディスクから読み出すようにすれば、ユーザの待ち時間を小さくすることができる。

もし、全てのPlayListファイルやClip Informationファイルを1つのファイルにまとめて記録すると、そのファイルサイズは非常に大きくなる。そのために、そのファイルの内容を変更して、それを記録するためにかかる時間は、個々のファイルを別々に分離して記録する場合に比べて、非常に大きくなる。本発明は、この問題を解決する。

上述した一連の処理は、ハードウェアにより実行させることもできるが、ソフトウェアにより実行させることもできる。一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが専用のハードウェアに組み込まれているコンピュータ、又は、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば汎用のパーソナルコンピュータ等に、記録媒体からインストールされる。

この記録媒体は、図108に示すように、コンピュータとは別に、ユーザにプログラムを提供するために配布される、プログラムが記録されている磁気ディスク221（フロッピディスクを含む）、光ディスク222（CD-ROM（Compact Disk-Read Only Memory）、DVD（Digital Versatile Disk）を含む）、光磁気ディスク223（MD（Mini-Disk）を含む）、若しくは半導体メモリ224等よりなるパッケージメディアにより構成されるだけでなく、コンピュータに予め組み込まれた状態でユーザに提供される、プログラムが記憶されているROM202や記憶部208が含まれるハードディスク等で構成される。

なお、本明細書において、媒体により提供されるプログラムを記述するステップは、記載された順序に従って、時系列的に行われる処理は勿論、必ずしも時系列的に処理されなくとも、並列的或いは個別に実行される処理をも含むものである。

また、本明細書において、システムとは、複数の装置により構成される装置全体を表すものである。

#### 産業上の利用可能性

以上の如く、本発明に係る情報処理装置及び方法、記録媒体のプログラム、ブ

ログラム、並びに記録媒体によれば、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方を、記録方法に応じて記録する。

本発明に係る情報処理装置及び方法、記録媒体のプログラム、並びにプログラムによれば、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている記録媒体からそれを再生し、出力を制御する。

また、本発明に係る情報処理装置及び方法、記録媒体のプログラム、プログラム、並びに第2の記録媒体によれば、主の再生パスを示す第1の情報と、前記主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を記録する。

本発明に係る情報処理装置及び方法、記録媒体のプログラム、並びにプログラムによれば、主の再生パスを示す第1の情報と、前記主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を記録媒体から再生し、それに基づいて出力を制御する。

したがって、何れの場合においても、高速再生が可能なAVストリームと不可能なAVストリームを、共通に管理することができる。また、アフターレコーディングが可能になる。

## 請求の範囲

1. AVストリームデータを記録媒体に記録する情報処理装置において、  
プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前  
記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、  
又は、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプ  
と、それに対応するトランスポートパケットの前記AVストリームデータ中のア  
ドレスとの対応関係を記述する第2のテーブルを生成する第1の生成手段と、  
記録方法に応じて前記第1のテーブル又は前記第2のテーブルの一方を選択す  
る選択手段と、

前記選択されたテーブルを前記AVストリームデータと共に前記記録媒体に記  
録する第1の記録手段とを有する情報処理装置。

2. 前記第1のテーブルは、EP\_mapであり、

前記第2のテーブルは、TU\_mapである請求の範囲第1項に記載の情報処理装置。

3. 前記選択手段は、ノンコグニザント記録の際には、前記第2のテーブルを  
選択する請求の範囲第1項に記載の情報処理装置。

4. 前記選択手段は、セルフエンコード記録の際には、前記第1のテーブルを  
選択する請求の範囲第1項に記載に記載の情報処理装置。

5. 前記選択手段は、コグニザント記録の際には、前記第1のテーブルを選択  
する請求の範囲第1項に記載に記載の情報処理装置。

6. 前記AVストリームデータの再生を指定する再生指定情報を生成する第2  
の生成手段と、

前記第2の生成手段により生成された前記再生指定情報を前記記録媒体に記録  
する第2の記録手段を更に有し、

前記再生指定情報は、前記AVストリームデータの再生区間の時間情報を、ブ  
レゼンテーションタイムベースで表現するか、又はアライバルタイムベースで表  
現するかを示す種別情報を含む請求の範囲第1項に記載に記載の情報処理装置。

7. 前記AVストリームデータと共に前記第1のテーブルが記録されている場

合、前記再生指定情報は、前記AVストリームデータの再生区間の時間情報を、プレゼンテーションタイムベースで表現し、

前記AVストリームデータと共に前記第2のテーブルが記録されている場合、前記再生指定情報は、前記AVストリームデータの再生区間の時間情報を、アライバルタイムベースで表現する請求の範囲第6項に記載に記載の情報処理装置。

8. AVストリームデータを記録媒体に記録する情報処理装置の情報処理方法において、

プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルを生成する生成ステップと、

記録方法に応じて前記第1のテーブル又は前記第2のテーブルの一方を選択する選択ステップと、

前記選択されたテーブルを前記AVストリームデータと共に前記記録媒体に記録する記録ステップとを含む情報処理方法。

9. AVストリームデータを記録媒体に記録する情報処理装置のプログラムにおいて、

プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルを生成する生成ステップと、

記録方法に応じて前記第1のテーブル又は前記第2のテーブルの一方を選択する選択ステップと、

前記選択されたテーブルを前記AVストリームデータと共に前記記録媒体に記録する記録ステップとを含むコンピュータが読み取り可能なプログラムが記録されている記録媒体。

10. AVストリームデータを記録媒体に記録する情報処理装置を制御するコ

ンピュータに、

プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランSPORTパケットの到着時刻に基づいたライバルタイムスタンプと、それに対応するトランSPORTパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルを生成する生成ステップと、

記録方法に応じて前記第1のテーブル又は前記第2のテーブルの一方を選択する選択ステップと、

前記選択されたテーブルを前記AVストリームデータと共に前記記録媒体に記録する記録ステップとを実行させるプログラム。

11. 記録媒体からAVストリームデータを再生する情報処理装置において、

プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランSPORTパケットの到着時刻に基づいたライバルタイムスタンプと、それに対応するトランSPORTパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている前記記録媒体から、前記第1のテーブル又は前記第2のテーブルの一方を再生する再生手段と、

再生された前記テーブルに基づいて、前記AVストリームデータの出力を制御する制御手段とを有する情報処理装置。

12. 記録媒体からAVストリームデータを再生する情報処理装置の情報処理方法において、

プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランSPORTパケットの到着時刻に基づいたライバルタイムスタンプと、それに対応するトランSPORTパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている前記記録媒体から、前記第1のテーブル又は前記第2のテーブルの一方を再生する再生ステップと、

再生された前記テーブルに基づいて、前記AVストリームデータの出力を制御する制御ステップとを含む情報処理方法。

13. 記録媒体からAVストリームデータを再生する情報処理装置のプログラムにおいて、

プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている前記記録媒体から、前記第1のテーブル又は前記第2のテーブルの一方を再生する再生ステップと、

再生された前記テーブルに基づいて、前記AVストリームデータの出力を制御する制御ステップとを含むコンピュータが読み取り可能なプログラムが記録されている記録媒体。

14. 記録媒体からAVストリームデータを再生する情報処理装置を制御するコンピュータに、

プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている前記記録媒体から、前記第1のテーブル又は前記第2のテーブルの一方を再生する再生ステップと、

再生された前記テーブルに基づいて、前記AVストリームデータの出力を制御する制御ステップとを実行させるプログラム。

15. AVストリームデータが記録されている記録媒体において、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに

対応するトランSPORTパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている記録媒体。

16. AVストリームデータを記録媒体に記録する情報処理装置において、主の再生パスを示す第1の情報と、前記主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を生成する生成手段と、

前記AVストリームデータと前記再生指定情報を前記記録媒体に記録する記録手段とを備える情報処理装置。

17. 前記副の再生パスは、オーディオデータのアフターレコーディング用のパスである請求の範囲第16項に記載の情報処理装置。

18. 前記第1の情報は、Main\_pathであり、

前記第2の情報は、Sub\_pathである請求の範囲第16項に記載の情報処理装置。

19. 前記第2の情報は、

前記副の再生パスのタイプを表すタイプ情報、

前記副の再生パスが参照する前記AVストリームのファイル名、

前記副の再生パスの前記AVストリームのイン点とアウト点、及び

前記再生パスのイン点が、前記主のパスの時間軸上で同期してスタートする前記主のパス上の時刻

を含む請求の範囲第16項に記載の情報処理装置。

20. AVストリームデータを記録媒体に記録する情報処理装置の情報処理方法において、

主の再生パスを示す第1の情報と、前記主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を生成する生成ステップと、

前記AVストリームデータと前記再生指定情報を前記記録媒体に記録する記録ステップとを含む情報処理方法。

21. AVストリームデータを記録媒体に記録する情報処理装置のプログラムにおいて、

主の再生パスを示す第1の情報と、前記主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を生成する生成ステップと、

前記AVストリームデータと前記再生指定情報を前記記録媒体に記録する記録ステップと

を含むコンピュータが読み取り可能なプログラムが記録されている記録媒体。

22. AVストリームデータを記録媒体に記録する情報処理装置を制御するコンピュータに、

主の再生パスを示す第1の情報と、前記主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を生成する生成ステップと、

前記AVストリームデータと前記再生指定情報を前記記録媒体に記録する記録ステップとを実行させるプログラム。

23. 記録媒体からAVストリームデータを再生する情報処理装置において、

主の再生パスを示す第1の情報と、前記主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を、前記記録媒体から再生する再生手段と、

再生された前記再生指定情報に基づいて、前記AVストリームデータの出力を制御する制御手段とを備える情報処理装置。

24. 記録媒体からAVストリームデータを再生する情報処理装置の情報処理方法において、

主の再生パスを示す第1の情報と、前記主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を、前記記録媒体から再生する再生ステップと、

再生された前記再生指定情報に基づいて、前記AVストリームデータの出力を制御する制御ステップとを含む情報処理方法。

25. 記録媒体からAVストリームデータを再生する情報処理装置のプログラムにおいて、

主の再生パスを示す第1の情報と、前記主の再生パスと同期して再生される副

の再生バスを示す第2の情報により構成される再生指定情報を、前記記録媒体から再生する再生ステップと、

再生された前記再生指定情報に基づいて、前記AVストリームデータの出力を制御する制御ステップとを含むコンピュータが読み取り可能なプログラムが記録されている記録媒体。

26. 記録媒体からAVストリームデータを再生する情報処理装置を制御するコンピュータに、

主の再生バスを示す第1の情報と、前記主の再生バスと同期して再生される副の再生バスを示す第2の情報により構成される再生指定情報を、前記記録媒体から再生する再生ステップと、

再生された前記再生指定情報に基づいて、前記AVストリームデータの出力を制御する制御ステップとを実行させるプログラム。

27. AVストリームデータが記録されている記録媒体において、主の再生バスを示す第1の情報と、前記主の再生バスと同期して再生される副の再生バスを示す第2の情報により構成される再生指定情報が記録されている記録媒体。

## 補正書の請求の範囲

[2001年8月16日(16.08.01)国際事務局受理:出願当初の請求の範囲  
1,3,4,5,6,7,8,9,10及び11は補正された;他の請求の範囲は変更なし。  
(3頁)]

1. (補正後) AVストリームデータを記録媒体に記録する情報処理装置において、

プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルを生成する制御部と、

記録方法に応じて選択された前記第1のテーブル又は前記第2のテーブルの一方を前記AVストリームデータと共に前記記録媒体に記録する記録部とを有する情報処理装置。

2. 前記第1のテーブルは、EP\_mapであり、

前記第2のテーブルは、TU\_mapである請求の範囲第1項に記載の情報処理装置。

3. (補正後) 前記制御部は、ノンコグニザント記録の際には、前記第2のテーブルを選択する請求の範囲第1項に記載の情報処理装置。

4. (補正後) 前記制御部は、セルフエンコード記録の際には、前記第1のテーブルを選択する請求の範囲第1項に記載の情報処理装置。

5. (補正後) 前記制御部は、コグニザント記録の際には、前記第1のテーブルを選択する請求の範囲第1項に記載の情報処理装置。

6. (補正後) 前記制御部は、さらに、前記第1テーブルまたは前記第2テーブルのどちらかが記録されているかを示す種別情報を生成し、前記記録部は、前記種別情報を記憶する請求の範囲第1項に記載の情報処理装置。

7. (補正後) 前記制御部は、前記AVストリームデータと共に前記第1のテーブルが記録されている場合、前記再生区間指定情報は、前記AVストリームデータの再生区間の時間情報をプレゼンテーションタイムベースで表現するとともに、前記AVストリームデータと共に前記第2のテーブルが記録されている場合、前記再生区間指定情報は、前記AVストリームデータの再生区間の時間情報をアラ

イバルタイムベースで表現するように制御する請求の範囲第1項に記載の情報処理装置。

8. (補正後) AVストリームデータを記録媒体に記録する情報処理装置の情報処理方法において、

プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランSPORTパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランSPORTパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルを生成する生成ステップと、

記録方法に応じて選択された前記第1のテーブル又は前記第2のテーブルの一方を前記AVストリームデータと共に前記記録媒体に記録する記録ステップとを含む情報処理方法。

9. (補正後) AVストリームデータを記録媒体に記録する情報処理装置のプログラムにおいて、

プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランSPORTパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランSPORTパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルを生成する生成ステップと、

記録方法に応じて選択された前記第1のテーブル又は前記第2のテーブルの一方を前記AVストリームデータと共に前記記録媒体に記録する記録ステップとを含むコンピュータが読み取り可能なプログラムが記録されている記録媒体。

10. (補正後) AVストリームデータを記録媒体に記録する情報処理装置を制御するコンピュータに、

プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランSPORTパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランSPORTパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルを生成する生成ステップと、

記録方法に応じて選択された前記第1のテーブル又は前記第2のテーブルの一方を前記AVストリームデータと共に前記記録媒体に記録する記録ステップとを実行させるプログラム。

11. (補正後) 記録媒体からAVストリームデータを再生する情報処理装置において、

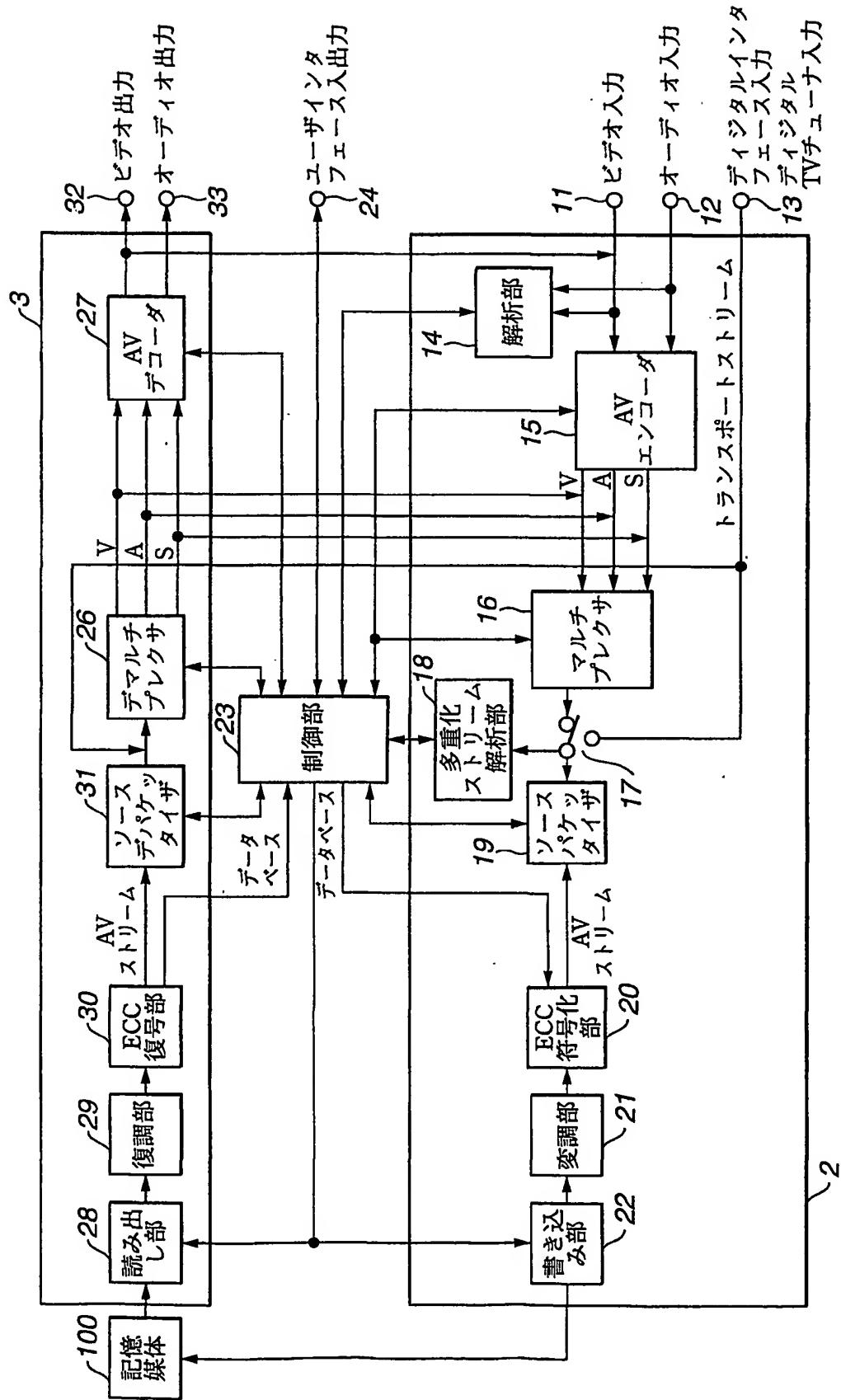
プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている前記記録媒体から、前記第1のテーブル又は前記第2のテーブルの一方を再生する再生部と、

再生された前記テーブルに基づいて、前記AVストリームデータの出力を制御する制御部とを有する情報処理装置。

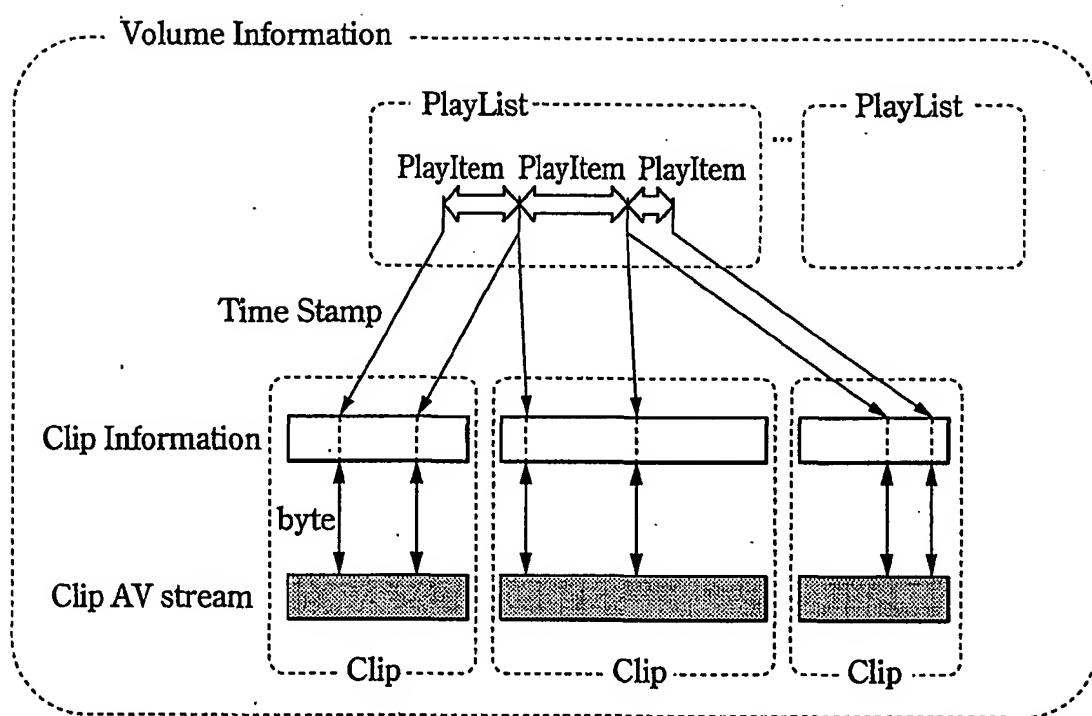
12. 記録媒体からAVストリームデータを再生する情報処理装置の情報処理方法において、

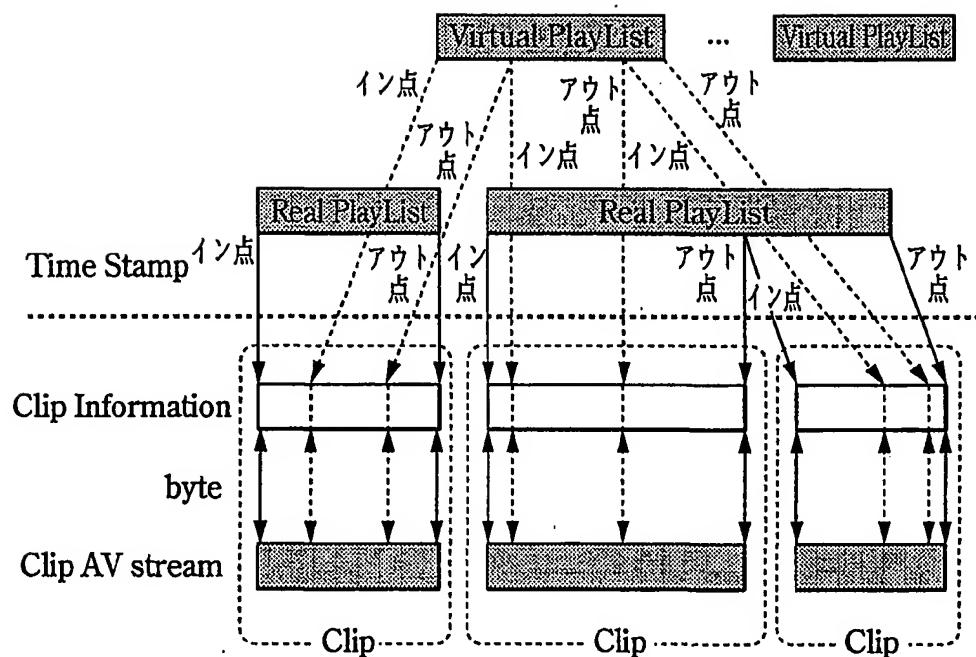
プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、又は、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている前記記録媒体から、前記第1のテーブル又は前記第2のテーブルの一方を再生する再生ステップと、

1/100

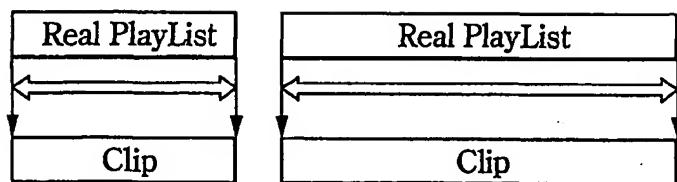
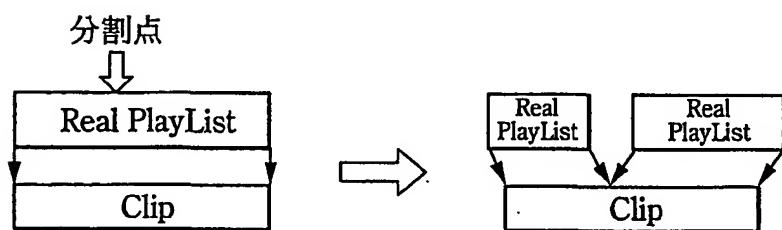
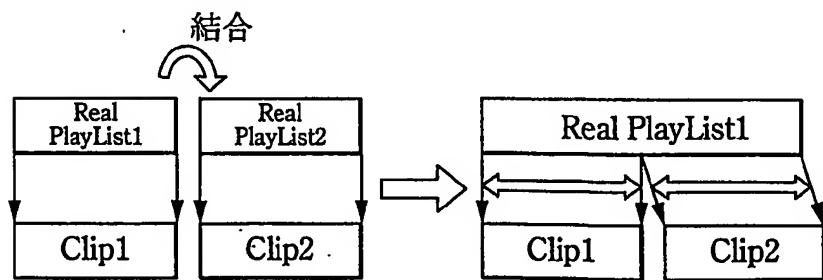


T  
E  
I  
G  
E

**FIG.2**

**FIG.3**

4/100

**FIG.4A****FIG.4B****FIG.4C**

5/100

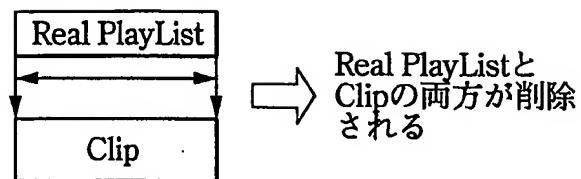


FIG.5A

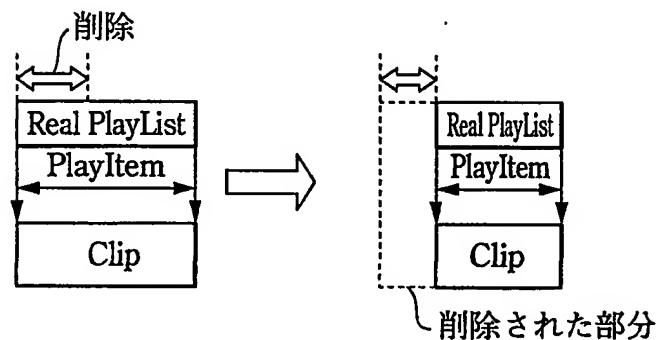


FIG.5B

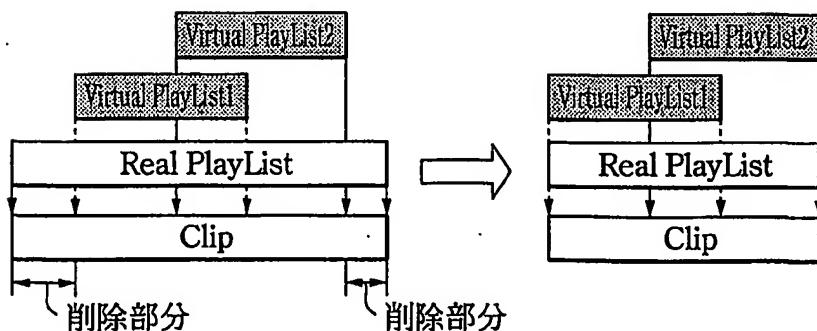
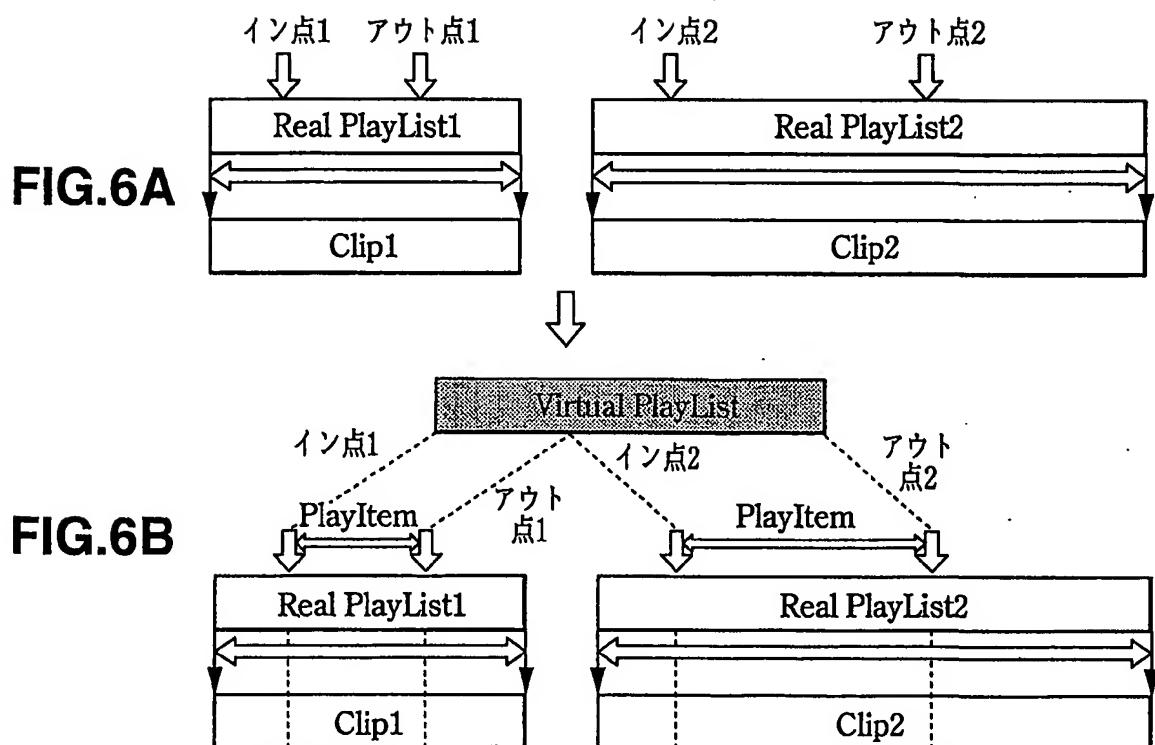


FIG.5C

6/100



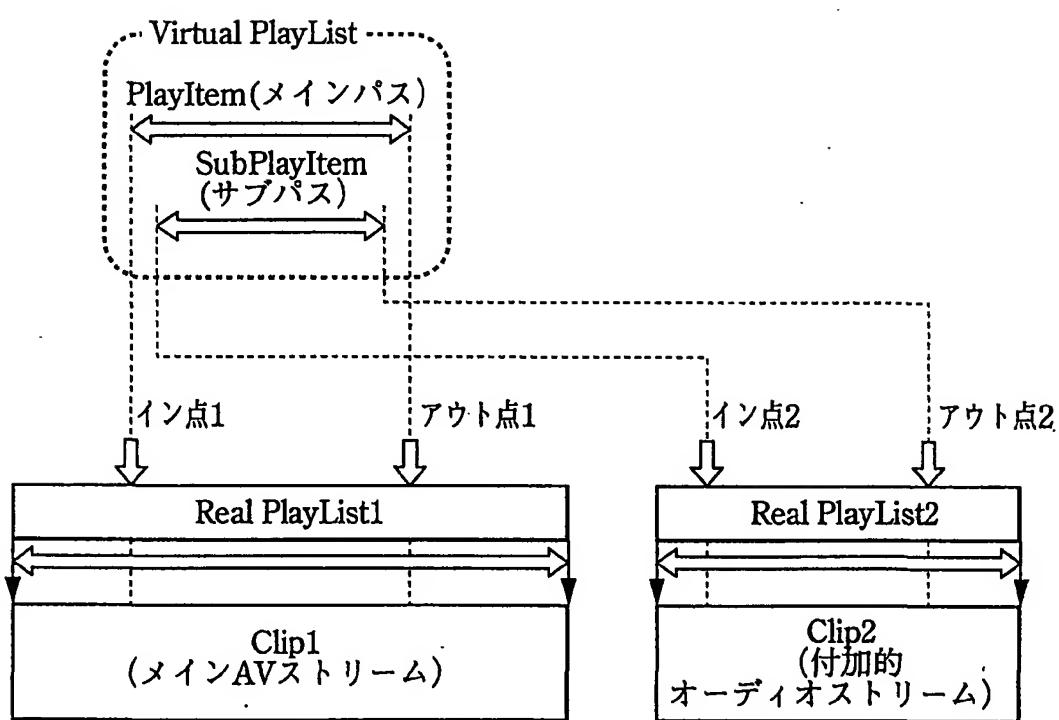


FIG.7

8/100

再生順序

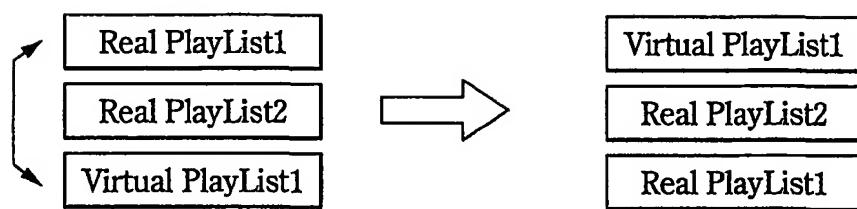


FIG.8

9/100

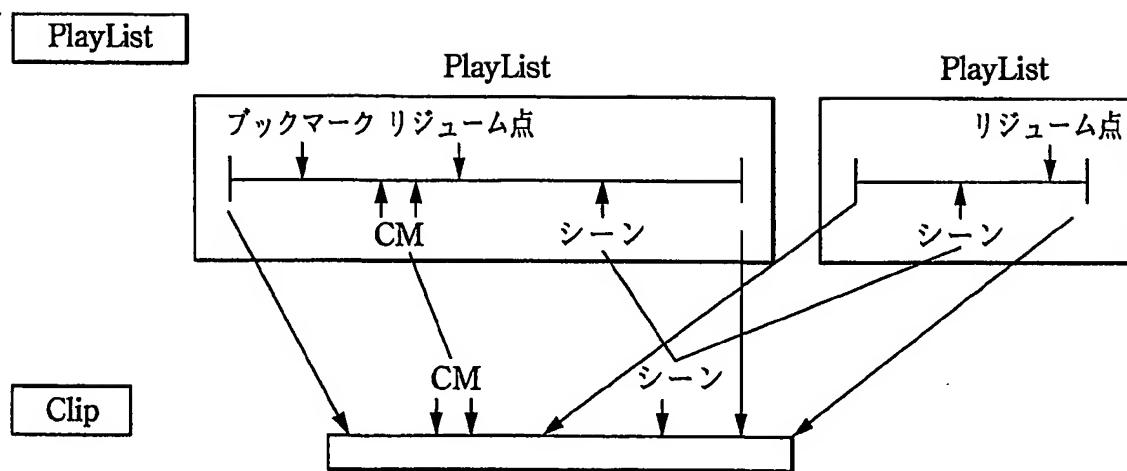


FIG.9

10/100

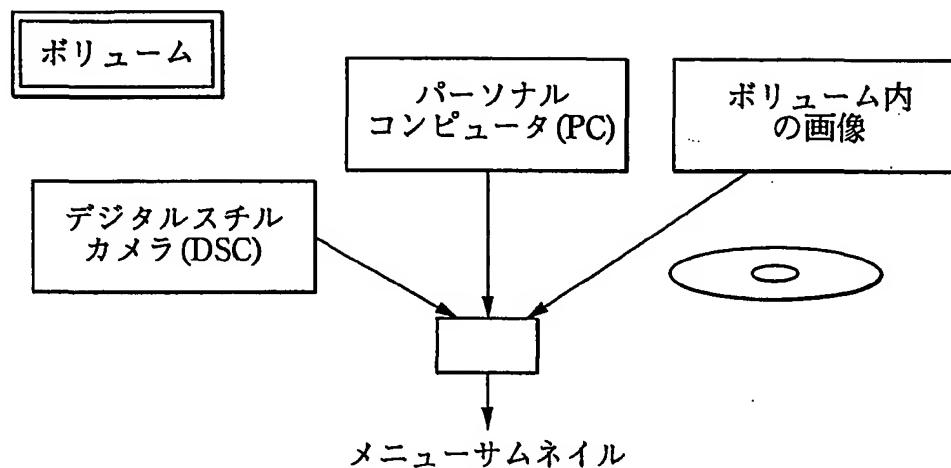


FIG.10

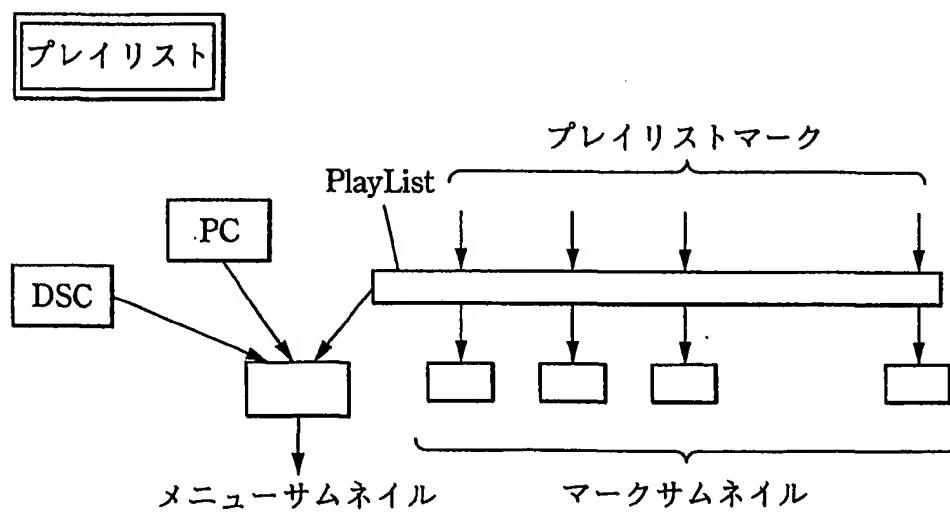


FIG.11

11/100

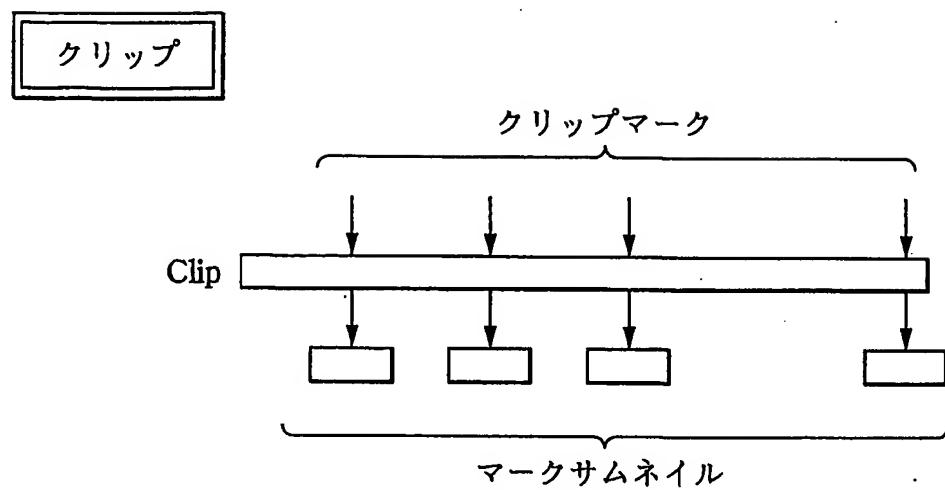


FIG.12

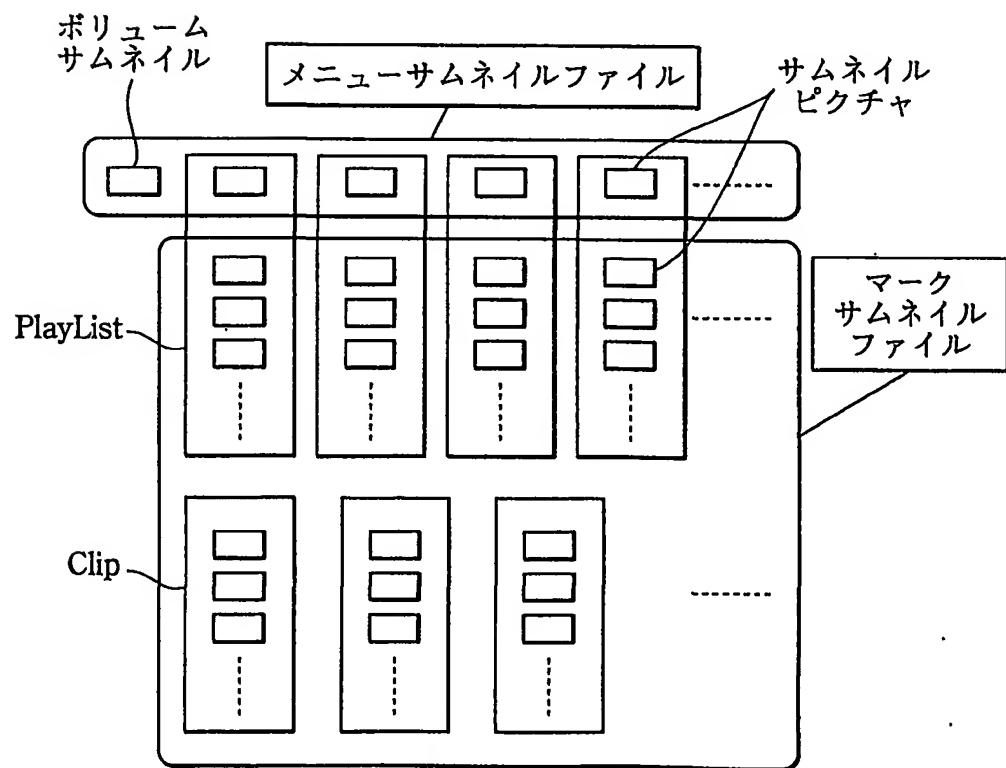


FIG.13

12/100

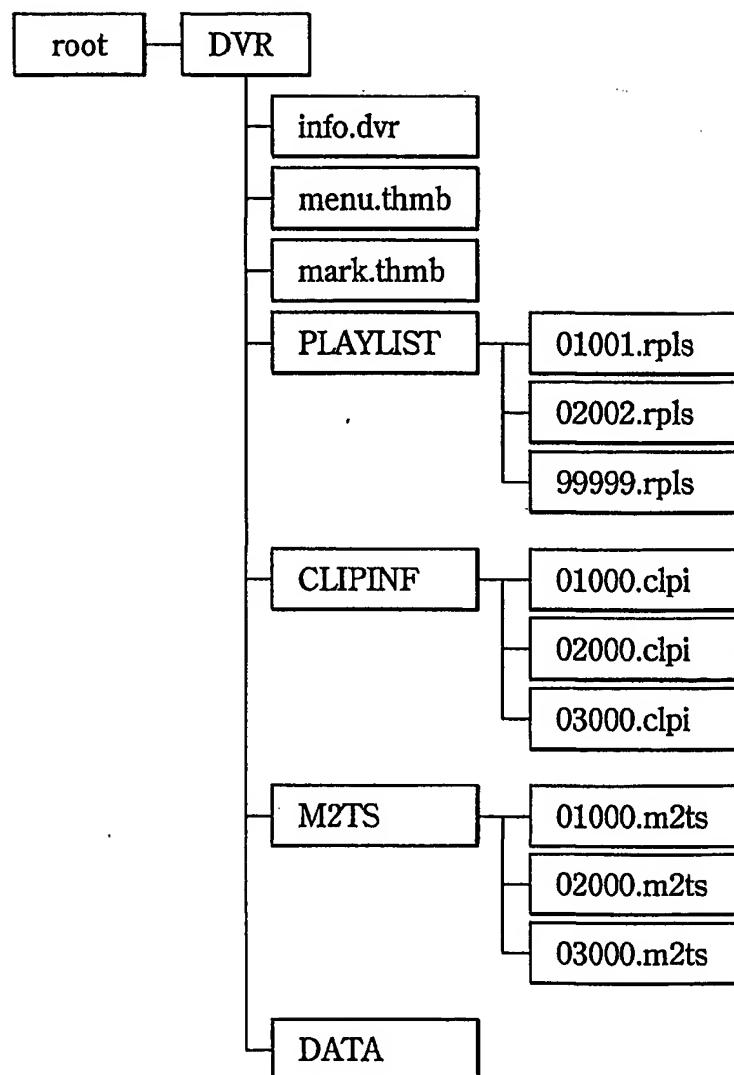


FIG.14

13/100

シンタクス	バイト数	略号
info.dvr {		
<b>TableOfPlayLists_Start_address</b>	32	uimsbf
<b>MakersPrivateData_Start_address</b>	32	uimsbf
<b>reserved</b>	192	bslbf
<b>DVRVolume()</b>		
<b>for (i=0;i&lt;N1;i++) {</b>		
<b>padding_word</b>	16	bslbf
<b>}</b>		
<b>TableOfPlayLists()</b>		
<b>for (i=0;i&lt;N2;i++) {</b>		
<b>padding_word</b>	16	bslbf
<b>}</b>		
<b>MakersPrivateData()</b>		
<b>}</b>		

FIG.15

14/100

シンタクス	バイト数	略号
DVRVolume()		
version_number	8*4	bslbf
length	32	uimsbf
ResumeVolume()		
UIAppInfoVolume()		
}		

FIG.16

15/100

シンタクス	バイト数	略号
ResumeVolume(){		
reserved	15	bslbf
valid_flag	1	bslbf
resume_PlayList_name	8*10	bslbf
}		

FIG.17

16/100

シンタクス	バイト数	略号
UIAppInfoVolume()		
character_set	8	bslbf
name_length	8	uimsbf
Volume_name	8*256	bslbf
reserved	15	bslbf
Volume_protect_flag	1	bslbf
PIN	8*4	bslbf
ref_thumbnail_index	16	uimsbf
reserved_for_future_use	256	bslbf
}		

FIG.18

**17/100**

値	キャラクタ文字符串化
0x00	Reserved
0x01	ISO/IEC 646 (ASCII)
0x02	ISO/IEC 10646-1 (Unicode)
0x03-0xff	Reserved

**FIG.19**

18/100

シンタクス	バイト数	略号
TableOfPlayLists()		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayLists	16	uimsbf
for (i=0; i<number_of_PlayLists; i++){		
PlayList_file_name	8*10	bslbf
}		
}		

FIG.20

19/100

シンタクス	バイト数	略号
TableOfPlayLists()		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayLists	16	uimsbf
for (i=0; i<number_of_PlayLists; i++){		
PlayList_file_name	8*10	bslbf
UIAppInfoPlayList()		
}		
}		

FIG.21

シンタクス	バイト数	略号
MakersPrivateData()		
version_number	8*4	bslbf
length	32	uimsbf
if (length !=0){		
mpd_blocks_start_address	32	uimsbf
number_of_maker_entries	16	uimsbf
mpd_block_size	16	uimsbf
number_of_mpd_blocks	16	uimsbf
reserved	16	bslbf
for (i=0; i<number_of_maker_entries; i++){		
maker_ID	16	uimsbf
maker_model_code	16	uimsbf
start_mpd_block_number	16	uimsbf
reserved	16	bslbf
mpd_length	32	uimsbf
}		
stuffing_bytes	8*2*L1	bslbf
for(j=0; j<number_of_mpd_blocks; j++){		
mpd_block	mpd_block_size*1024*8	
}		
}		
}		

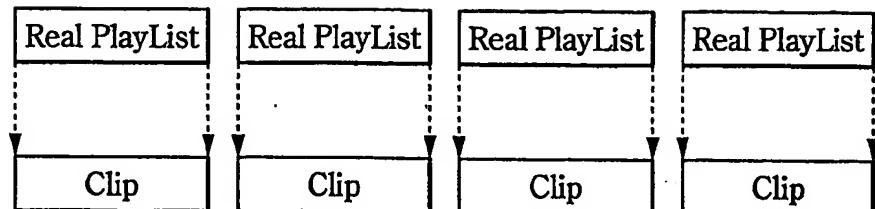
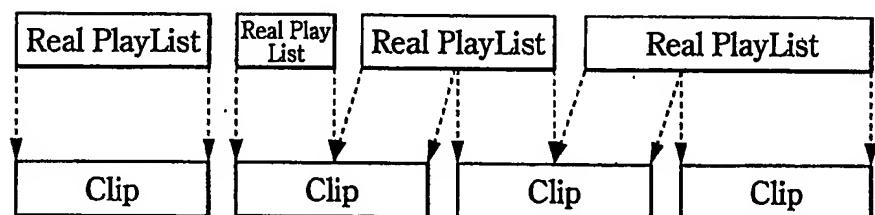
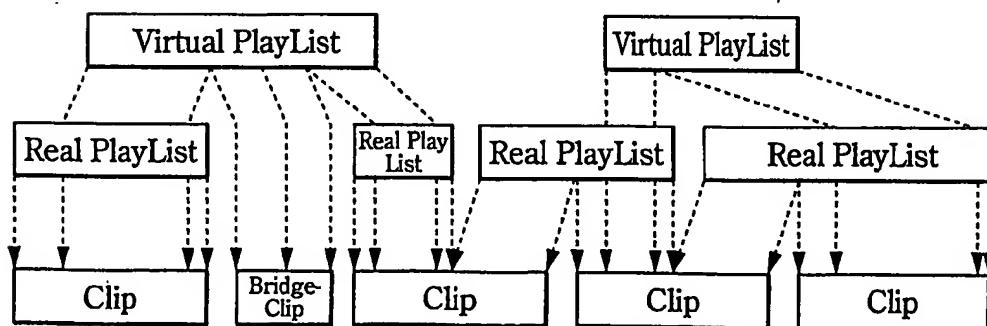
FIG.22

21/100

シンタクス	バイト数	略号
xxxxx.rpls / yyyy.vpls {		
PlayListMark_Start_address	32	uimsbf
MakersPrivateData_Start_address	32	uimsbf
reserved	192	bslbf
PlayList()		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
PlayListMark()		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

FIG.23

22/100

**FIG.24A****FIG.24B****FIG.24C**

シンタクス	バイト数	略号
PlayList()		
version_number	8*4	bslbf
length	32	uimsbf
PlayList_type	8	uimsbf
CPI_type	1	bslbf
reserved	7	bslbf
UIAppInfoPlayList()		
number_of_PlayItems // main path	16	uimsbf
if (<Virtual PlayList>){		
number_of_SubPlayItems // sub path	16	uimsbf
}else{		
reserved	16	bslbf
}		
for (PlayItem_id=0;		
PlayItem_id<number_of_PlayItems;		
PlayItem_id++){		
PlayItem() //main path		
}		
if (<Virtual PlayList>){		
if (CPI_type==0 && PlayList_type==0){		
for (i=0; i<number_of_SubPlayItems; i++)		
SubPlayItem() //sub path		
}		
}		
}		

FIG.25